microcomputer components

MOTOROLA Semiconductors

contents

Page

Chapter 5 — Bipolar 4-Bit Slice Microprocessor Families

TTL

MECL

motorola microcomputer components

What you should consider before you rush into — microcomputer design

What's so important about the microprocessor that it should have been catapulted into one of the most significant developments in the electronics field over the last decade? Well, certainly not that microprocessors, themselves, represent anything new in the way of circuit implementation. Computer designers had been using them as integral parts of large computers long before they became the buzz-word of the industry. Rather, it's the fact that technological advances into large-scale integration have made the microprocessor the most effective, most reliable, simplest, and *least expensive* way of accomplishing complex electronic functions today. It's not surprising, therefore, that the microprocessor, or in a larger sense, the microcomputer, is dominating the design thinking for most equipment currently in the planning stage.

The microcomputer is such a powerful performer that the vast majority of all possible applications probably could be served by any one of the dozens of different models available. Yet, for each application there's probably one microcomputer system that serves the purpose *best*. And so, before entrusting your design to any specific processor (or supplier) we offer the following four considerations for your investigation:

Choice

Despite occasional claims to the contrary, there's no such thing as a truly *universal* circuit—not from the standpoint of cost-effectiveness. While our M6800 microcomputer system comes as close as any other to serving the bulk of all potential applications, it's "overqualified" for some, and "underqualified" for others. That's why Motorola manufactures a variety of microcomputer systems:

The M6800 Family—the most pervasive of the general-purpose MPU systems,
The MC3870—a low-cost, single-chip microcomputer for dedicated applications,
The MC141000—a CMOS alternative to the above, for lowest possible power dissipation,
The bipolar M2900 and M10800 systems—for highest speed.

Moreover, each of these has at least one viable alternate source, so that your manufacturing requirements can not easily be compromised.

In addition to components for microcomputer systems, Motorola supplies an extensive line of micromodules—assembled subsystems—for those manufacturers who wish to begin their equipment designs at a higher level.

Support

The key to the successful development of a dedicated MPU system and, ultimately, to manufacture and service the system, is an umbrella of support equipment. The Motorola microcomputer product families are complemented with one of the industry's most pervasive arrays of user-oriented development aids, test equipment and support literature. The support hardware system is modularized to permit purchase of just those components required for the complexity of the system to be designed. An extensive software library, including a proven library of user-developed

MOTOROLA MICROPROCESSING MANUFACTURING FACILITIES

AUSTIN, TEXAS
Manufacturing facility for all Motorola MOS products

MESA, ARIZONA
Bipolar processors and other MPU-related bipolar integrated circuits

programs, is also available. And Motorola maintains a nationwide network of Field Applications Engineers to assist customers with microcomputer design problems.

Commitment

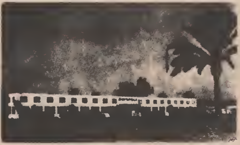
Chances are your first MPU purchase will not be your last. That's why a primary consideration in the choice of an MPU line on which to hang your designs should be the manufacturer's commitment to the expansion of that specific line—expansion of hardware to keep up with the state of the art while maintaining software compatibility with the existing system.

Motorola's commitment to the M6800 line is manifest not only by the introduction of a second-generation MPU (the MC6802), but by a literal explosion of scheduled peripheral products encompassing over 100 new type numbers in 1977 alone. All are bus-compatible with the original system and utilize the original instruction set. This dedication assures the safety of your software investment even as you switch to increasingly cost-effective hardware for future design and production.

Diversification

Even the most complex microprocessor system (or single-chip microcomputer) isn't a complete system. All need additional components of one kind or another to perform an equipment function. More memory, perhaps, for additional storage capacity; interface circuits to match various peripherals; power devices to drive external equipment. Motorola's product line extends far beyond MPU device lines. As a world leader in solid-state products, we supply devices in almost every semiconductor category—from ICs to discretes; from digital to linear; from MOS to bipolar. This proven solid-state capability is your further insurance of total-product support—today, tomorrow, and in the years ahead.

May we help you?

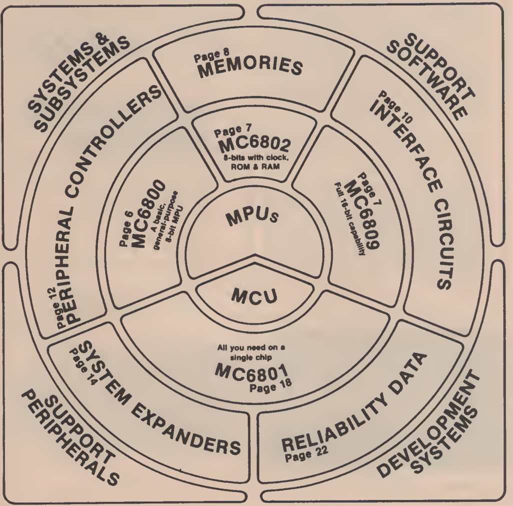


PHOENIX, ARIZONA
Microsystems and support products (56th Street Facility).

table of contents

For a maximum versatility —

the M6800 microcomputer family . . .


SYSTEMS & SUBSYSTEMS
SUPPORT SOFTWARE
SUPPORT PERIPHERALS
DEVELOPMENT SYSTEMS
Page 8 MEMORIES
Page 10 INTERFACE CIRCUITS
Page 12 PERIPHERAL CONTROLLERS
SYSTEM EXPANDERS Page 14
RELIABILITY DATA Page 22
Page 7 MC6802 8-bits with clock, ROM & RAM
Page 6 MC6800 A basic, general-purpose 8-bit MPU
Page 7 MC6809 Full 16-bit capability
All you need on a single chip MC6801 Page 18
MPUs
MCU

at a glance . . .

Inherent in the formula for successful MPU-based designs is the selection of the most cost-effective processor family from among the many systems available today. The M6800 family ranks high in meeting the requirements.

Its NMOS LSI architecture offers bus transfer rates up to 2 MHz, and 8- and 16-bit processing capability.

Its powerful instruction set minimizes memory requirements and enhances system throughout.

The progressive complexity of its basic MPU/MCU building blocks permits system design flexibility that yields cost-effectiveness for any potential applications

But there are more considerations to the selection of the best microcomputer system than just technical capability. When your production is as heavily dependent on the availability of a set of components as dictated by a commitment to a specific MPU family, a constant and reliable source of supply is of paramount importance. So is the continuing flow of new and related products that guards against system obsolescence.

Motorola is dedicated to the continual expansion of the M6800 system with new products, new designs and expanded peripherals—all tailored to increase the scope and value of your investment.

M6800 LINE FEATURES

For System Design Ease:

- Powerful, variable-length instructions reduce programming complexity and development time.
- 65K memory address capability encompasses the largest program requirements likely to be encountered in microcomputers.
- Single 5-volt power supply operation and bus organization simplifies system design.

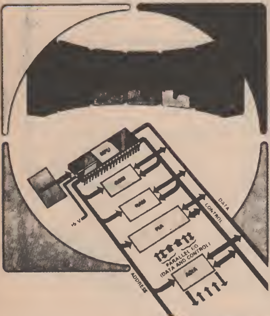
For Maximum Throughput:

- Bus transfer rates to 2 MHz provide high-speed operation.
- Automatic data stacking during interrupts reduces programming complexity.
- 3-state output.
- Vectored restart

M6800
the MPU

Whether a microcomputer consists of a totally integrated single chip, or is composed of a number of interactive LSI chips, *the microprocessing unit (MPU) is the central control system that determines the eventual application for which the system is best suited.* Its architecture contains the complex routines that permit the system to respond correctly to each of the different "instructions" associated with a particular system. It controls the flow of signals into and out of the computer, routing each to its proper destination in the required sequence to perform an end function.

The M6800 Family currently includes two standard 8-bit MPUs, with a third, a 16-bit unit, scheduled to join the lineup during 1978. And for maximum on-chip power, a complete single-chip microcomputer will join the Family soon (see Page 18).



The Original Lineup

THE MC6800

This microprocessor was the first of the M6800 MPU Family and still remains a highly cost-effective processor for a great many process-control and data-communications applications. Seventy-two powerful instructions and six different addressing modes give it unexcelled capability and a full range of compatible peripheral chips offer the widest possible latitude in system implementation. After years of field experience, the MC6800 has earned an enviable reputation as one of the easiest to use processors available because:

Its bus organized architecture reduces component count and simplifies interconnection;

Its single 5-V supply requirement reduces system complexity and cost;

Its 16-bit address system permits selective addressing of more than 65,000 memory locations;

Its inherent design treats each peripheral as a memory location, thereby reducing programming complexity.

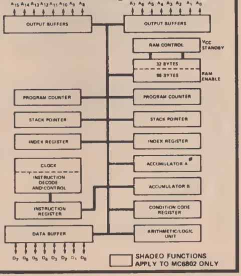
Moreover, to tailor the system to your specific needs at the lowest cost, the MC6800 (and its peripherals) is available in two different packages, three different temperature ranges and three speed ranges, as follows:

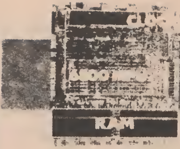
Selection

Temperature Range	Frequency Limit		
	1 MHz	1.5 MHz	2 MHz
0 to 70°C	MC6800P/L	MC68A00P/L	MC68B00P/L
40 to 85°C	MC6800CL		
55 to 125°C	*MC6800CQCS		

P suffix - Plastic Package   L suffix - Ceramic Package

MC6800/6802 — Programmer's Model





MC6802

Take the basic MC6800 MPU, add an on-chip clock and 128 bytes of RAM, and you essentially have the second generation M6800 MPU chip—the MC6802. This versatile processor has all the attributes of the basic unit.

—It is fully compatible with all the peripherals, features the same MPU architecture and capabilities, and works with the same instruction set—

But it reduces the component count of a minimum microcomputer system to only *two*, compared with a minimum of four with the earlier MPU.

The built-in clock operates at a maximum frequency of 1 MHz but, thoughtfully, the chip designers have added an on-chip divide-by-four circuit to permit the use of an external 4-MHz crystal in lieu of a far more expensive 1-MHz crystal. In addition the first 32 bytes of the built-in RAM may be operated in a low-power mode, from an external power source, to prevent the loss of information during a power-down situation.

Utilizing this MPU, a minimum microcomputer system consists of:

1 MC6802 MPU
1 MC6846 ROM-I/O-Timer Unit (Page 10)

Of course, the system is expandable to any requirement with the adapters, expanders and other peripheral chips that are a part of the M6800 Family.

The MC6802 is available in both ceramic (suffix L) and plastic (suffix P) package.

INTRODUCING THE MC6809 MICROPROCESSOR

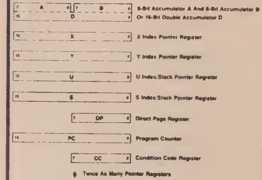
(Coming Soon)

Today, there's a lot of controversy about where a microcomputer turns into a "mini". While a number of benchmarks have been suggested, it is generally conceded that 16-bit processing capability constitutes a minimum "mini" requirement.

Motorola is a microcomputer manufacturer, but the soon-to-be-announced MC6809 Microprocessor at least borders on minicomputer capabilities.

It has 16-bit capability with 50-percent more throughput than the MC6800. It operates at 2 MHz, adds 16 new addressing modes, utilizes an expanded instruction set with high-level language capability, and features a host of other refinements that add functional expansion to, while maintaining compatibility with, the M6800 Microcomputer Components Family.

MC6809 Programming Model

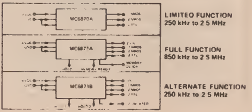


7 A 0 7 B 0
15 D 0
8-Bit Accumulator A And 8-Bit Accumulator B
Or 16-Bit Double Accumulator D
15 X 0
X Index Pointer Register
15 Y 0
Y Index Pointer Register
15 U 0
U Index/Stack Pointer Register
15 S 0
S Index/Stack Pointer Register
7 DP 0
Direct Page Register
15 PC 0
Program Counter
7 CC 0
Condition Code Register
● Twice As Many Pointer Registers

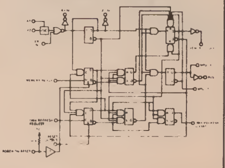
M6800 CLOCKS

All M6800-based systems operate with two non-overlapping clock phases, ϕ1 and ϕ2. A variety of clock modules is available for use with the M6800 MPU (other MPUs have built-in clock). Variations include one monolithic and three hybrid versions offering a number of system design options.

THE HYBRID



THE MONOLITHIC MC6875



interface circuits for

peripherals

Simple Microcomputers, or those dedicated to one specific application, conceivably could have all the required circuitry on a single chip. General-purpose microcomputers, and those intended for complex system design, do not. The reason—the design of a chip with the memory and interface circuitry compatible with *all* possible end-use applications would make it cost-effective for *none*.

The M6800 system was conceived and designed to encompass an array of LSI components which, in various combinations, provide low-cost solutions to most eventual microcomputer applications. The choice of microprocessor chips, Page 6, together with the selection of the right memory, Page 8, and the most suitable peripheral interface circuits described here, often results in the most effective and least expensive system that can be configured for most applications.

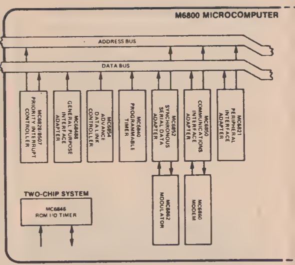
MC6846* — ROM-I/O-Timer

Highly efficient interface chip contains 2048 bytes of ROM, togethar with a 16-bit programmable timer-counter and an 8-bit bidirectional data port for peripheral interface. In conjunction with MC6802 MPU, it constitutes a versatile 2-chip microcomputer system Compatibility with other M6800 interface and peripheral circuits permits system expansion to any required additional complexity at low cost.

The built-in ROM provides read-only storage for a minimum microcomputer system and is mask-programmable to tha usar's specifications Tha timer may be programmed to count events, measure frequencies and time intervals, generate square waves, etc The I/O port is under software control and includas two "handshake" control lines for asynchronous interface with peripherals.

MC6821* — Peripheral Interface Adapter (PIA)

This parallel orianted periperal interface circuit is one of the most important intarface circuits available Contains two I/O circuit blocks, each capable of controlling an independant 8-bit peripheral data bus Multiple PIAs can be used with a single system and salactively addressed by maans of Chip-Salect inputs Each peripheral data line can be programmed to act as an input or output and each of four control/interrupt lines can operate in ona of several control modes.


M6800 MICROCOMPUTER
ADDRESS BUS
DATA BUS
MC6828/8507 PRIORITY INTERRUPT CONTROLLER
MC68488 GENERAL PURPOSE INTERFACE ADAPTER
MC6854 ADVANCE DATA LINK CONTROLLER
MC6840 PROGRAMMABLE TIMER
MC6852 SYNCHRONOUS SERIAL DATA ADAPTER
MC6850 COMMUNICATIONS INTERFACE ADAPTER
MC6821 PERIPHERAL INTERFACE ADAPTER
TWO-CHIP SYSTEM
MC6846 ROM I/O TIMER
MC6862 MODULATOR
MC6860 MODEM

MC6828/8507 — Priority Interrupt Controller (PIC)

This bipolar device is used to add prioritized responses to inputs of microprocessor systems. The performance has been optimized for the M6800 system, but will serve to eliminate input polling routines from any procassor system.

With the PIC, each interrupting davice is assigned a unique ROM location which contains the starting addrass of the appropriata service routine. Aftar the MPU detects and rasponds to an intarrupt, tha PIC directs the MPU to tha proper memory location.

MC6840* — Programmable Timer

This component is designed to provide variable system time intarvals. It has three 16-bit binary counters, three corresponding control registers and a status register. The countars are under software control and may be used to cause system interrupts and/or generata out-put signals. Tha MC6840 may be utilized for frequancy measurements, event counting, intarval maasuring and similar tasks.

MC68488* — General-Purpose Interface Adapter

Tha MC68488 GPIA interfaces between the IEEE488 standard instrument bus and the M6800 System. With it, many instruments may be interconnected and remotaly and automatically controlled or programmed. Data may ba taken from, sent to, or transfarred between instruments.

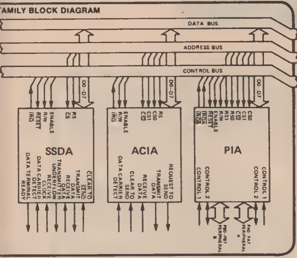
The MC68488 will automatically handle all handshake protocol needed on the instrument bus.

*Available in package configurations and tamperatura and frequency ranges to match those of the MC6800 MPU described on Page 6

communications and instrumentation

In many applications, input data to a computer comes from program sources that are wired directly to the computer inputs; in others the data is derived from remotely located sources and transmitted to the computer by means of telephone lines. Remote data communications requires additional peripheral equipment—to establish contact; to convert digital signal levels into corresponding transmittable data; to assemble serially transmitted data pulses into byte-sized parallel inputs to the computer (or vice-versa).

The M6800 Family contains a number of compatible LSI components that make the development of communications interface equipment quick, easy and relatively inexpensive.



MC6860* — 0–600 bps Digital Modem

The MC6860 is a MOS subsystem designed to be integrated into a wide range of equipment utilizing serial data communications, including stand-alone modems, data-storage devices, remote-data communications terminals and I/O interfaces for minicomputers.

The modem provides the necessary modulation, demodulation and supervisory control functions to implement a serial data communications link, over a voice grade channel, utilizing frequency shift keying (FSK) at bit rates up to 600 bps.

The modem is compatible with the M6800 Microcomputer Family, interfacing directly with the Asynchronous Communications Interface Adapter to provide low-speed data communications capability.

MC6850* — Asynchronous Communications Interface Adapter (ACIA)

This circuit provides the data formatting and control to interface serial asynchronous data communications information to bus-organized systems.

The parallel data of the bus system is serially transmitted and received by the asynchronous data interface, with proper formatting and error checking. The functional configuration of the ACIA is programmed via the data bus during system initialization. A programmable Control Register provides variable word lengths, clock division ratios, transmit control, receive control, and interrupt control. Three control lines allow the ACIA to interface directly with the MC6860 Digital Modem.

MC6854* — Advanced Data Link Controller (ADLC)

The MC6854 ADLC performs the complex MPU/data communication link function for the "Advanced Data Communication Control Procedure" (ADCCP), High-Level Data Link Control (HLDC) and Synchronous Data Link Control (SDLC) standards

In a bit-oriented data communication system the data is transmitted and received in a synchronous serial form.

The serial data stream must be converted into parallel, analyzed, and stored (for use by the MPU) in order for data link management to be accomplished. Similarly, parallel data from the MPU system must be serialized with the appropriate frame control information in order to conform to the bit-oriented protocol standards. The Advanced Data Link Controller (ADLC) provides these functions.

MC6852* — Synchronous Serial Data Adapter (SSDA)

Provides interface between the M6800 MPU system and synchronous data terminals such as floppy disk equipment, cassette or cartridge tape controllers, numerical control systems and other systems requiring movement of data blocks. Operates at speeds up to 600 kbps.

MC6862 — Digital Modulator

Offers the necessary modulation and control functions to implement a serial data communications link over voice-grade channels at bit rates of 1200 and 2400 bps.

*Available in package configurations and temperature and frequency ranges to match those of the MC6800 MPU described on Page 6

New M6800

LSI peripheral controllers

On the one hand there are the microcomputers, and on the other there are the peripherals. Each peripheral has different needs, both functionally and electrically, and, therefore, demands a different interface circuit to adapt it to a specific microcomputer design.

The MC6821 Peripheral Interface Adapter on Page 10 permits first-order peripheral selection and I/O control, but it doesn't provide the complex functional control required by each unique computer peripheral.

Normally, functional peripheral control requires a board-full of SSI/MSI circuits. The LSI circuits described here reduce this requirement to a simple, convenient and relatively inexpensive single package.

MC6843 — Floppy Disk Controller

This 40-pin LSI circuit performs the complex MPU/Floppy Disk interface function. It contains twelve accessible and three nonaccessible internal registers which, together with a Micro-Controller/ROM network, form the communications link between the M6800 MPUs and a wide range of disk drives. Multiple disk drives can be controlled with the addition of external multiplexing rather than additional controllers.

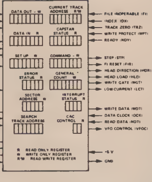
General Description

The three nonaccessible registers provide serial-to-parallel and parallel-to-serial conversions as well as Data-Clock pattern generation and detection. The disk operation is monitored by the MPU via the three status registers. Separate registers provide for Track and Section Address Information.

The Setup Register serves two purposes. One section allows generation of a programmable delay corresponding to the Seek Time of the drive in use. The remaining section provides a programmable settling time delay.

The General Count Register provides the new track number for SEK and STZ commands, and a second count for multi-sector read/write.

One bit in the Command Register selects either Program Control or Direct Memory Access. The remaining bits in the Command Register direct the internal Micro-Control Unit to perform either a micro or macro command. A set of 10 macro commands govern program operation.



Programming Model of MC6843 Floppy Disk Controller

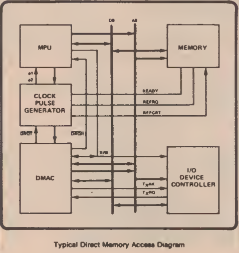
MC6844 — Direct Memory Access Controller

This DMAC works with an M6800 MPU Clock Pulse Generetor and an I/O Peripherel Controller, such as the units described here, to facilitete direct eccess to the computer memory by the peripheral, thus bypassing MPU interactive time delay.

General Description

The MC6844 is operable in three modes: HALT Burst, Cycle Steal and TSC Steal. In the Burst Mode, the MPU is halted by the first transfer request (TxRQ) input and is restarted when the Byte Count Register (BCR) is zero. Each data transfer is synchronized by a pulse input of TxRQ. In the Cycle Steal Mode, the MPU is helted by eech TxRQ end is resterted efter eech one byte of dete trensferred. In the TSC Steel Mode, DMAC uses the three-stete control function of the MPU to control the system bus. One byte of deta is trensferred during each DMA cycle.

The DMAC has four channels. A Priority Control Register determines which of the channels is enabled. While data is being transferred on one channel, the other chennels ere inhiblted. When one channel completes transferring, the next will become valid for DMA transfer. The PCR elso utilizes a Rotate Control bit. Priority of DMA transfer is normally fixed in sequentiel order. The highest priority is in #0 Chennel and the lowest is in #3. When this bit is in high level, chennel priority is rotated such that the just-serviced channel hes the lowest priority in the next DMA transfer.



Typical Direct Memory Access Diagram

MC6845 — CRT Controller

This single-chip Controller provides the complex interface between a cathode-ray terminal and a microprocessor of the M6800 Family. It is designed to simpllfy the development and production of equipment such as intelligent terminals, word processing and information displey devices.

General Description

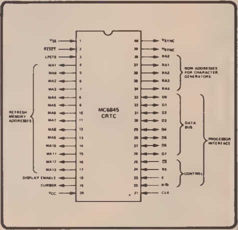
The CRTC consists of the horizontal and vertical counting circuits, a display address generator, a cursor register and comparator, and a light-pen register.

The horizontal and vertical counting circuits generate the signals: Blank, HSYNC, VSYNC, and R0–R4. R0–R4 are row count signals to the external charecter generetor ROM. The numbers of horizontal cherecters per raster, rasters per charecter line, character lines per screen and horizontel and vertical SYNC position are progremmeble by the MPU.

With 14 address lines from the CRTC to the display memory, over 16K of memory may be randomly addressed for display. The CRT may be scrolled or paged through the entire display memory under MPU control.

A light pen strobe input signel allows capture of refresh address in en internal light-pen register.

The cursor control register determines the cursor locetion on the screen. The cursor format can be programmed for fast-blink, slow-blink, or non-blink appeerence, with programmeble size.



Pin Assignment for MC6845

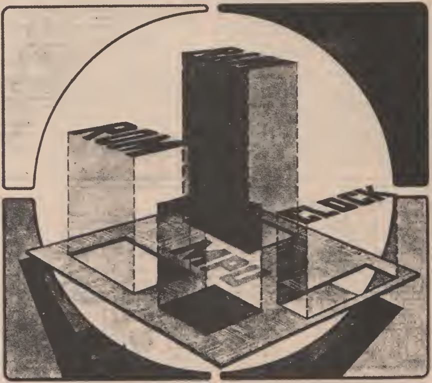
all-on-one-chip
microcomputers

Utilization of large-scale integrated circuits always involves a series of compromises. On the one hand, the more circuitry that is incorporated on a single chip, the more limited is the system designer's control over the exact behavior of the ultimate system. On the other hand, the more complex the chip, the fewer the components needed for system implementation and, concurrently, the simpler the design and the lower the cost.

Applied to microcomputers, the previously described M6800 Family of LSI components gives the designer control over system architecture and functional operation while, at the same time, offering large enough building blocks for simplified system designs at low cost. Yet, there are numerous microcomputer applications where the end requirement permits processing of all necessary functions—MPU, Memory, I/O—into a single chip of managable size. For such applications the single-chip microcomputer results in the most cost-effective final system.

You don't just order a single-chip microcomputer; you have it built for you. Since your dedicated program must be incorporated in the read-only memory on the chip, your order constitutes, in effect, a custom order. Yet, because custom programming can be done in the final metallization stage of chip processing, such processing is relatively inexpensive.

Motorola now supplies two single-chip microcomputers of varying design with still another in the design cycle. These components, described on the following pages, merit serious consideration.



For dedicated applications . . .

single chip microcomputers

Any application that can be formatted within the capacity of an on-chip ROM is a likely candidate for one of the single-chip microcomputers described here. The customer develops and tests his proprietary source program and sends it to Motorola for proper processing of the final chip. From receipt of the source program to delivery of prototype product takes approximately 8 weeks.

The MC3870 8-Bit MOS

Take a powarful Arithmatic Logic Unit (ALU);
Plus 2048 bytes of Read-Only Mamory (ROM) and 64 bytes of scratchpad RAM;
Add four ports of TTL-compatibla Input/Output; a programmable binary timer capabla of operating in the Interval Timer mode, the Pulse-Width Measurement mode and the Evant Countar Mode; a built-in clock with intarnal or extarnal timing capability;
Make it completely compatible with the extensive software library of tha popular F8 microcomputer . . .
And *you'va got as versatile* a *single-chip microcomputar as modarn technology permits.*

Tha Motorola MC3870 is a complata 8-bit MOS microcomputar utilizing ion-implanted, N-channel, silicon-gate technology and offars maximum cost-effectiveness for a wida range of control and logic-raplacement applications. It is simple to implamant (requiring only a single +5-volt ±10% power supply) and power saving in operation (requires only 275 mW, typical). Seventy-six instructions, tha entire instruction set of tha F8 multi-chip family, impart to tha MC3870 a high degree of functional varsatility.

Functional Pin Description

P0-0 to P0-7, P1-0 to P1-7, P4-0 to P4-7, and P5-0 to P5-7 are 32 lines which can be individually used as aither TTL-compatible inputs or as latched outputs.

STROBE is a ready strobe associated with I/O Port 4. This pin, which is normally high, provides a single low pulse aftar valid data is presented on the P4-0 to P4-7 pins during an output instruction.

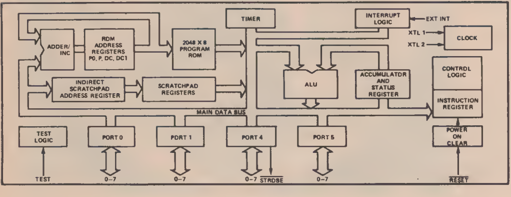
RESET may be used to externaily reset tha 3870. When pulled low, tha 3870 will reset. When then allowed to go high, the 3870 will begin program axecution at program location H "000".

EXT INT is the axternal intarrupt input. Its activa state is software programmabla. This input is also used in conjunction with tha timer for pulse-width measurement and event counting.

XTL1 and XTL2 are the time base inputs to which a crystal (1 to 4 MHz), LC network, RC network, or an axternal single-phase clock may be connected. If timing is not critical, the 3870 will operata from its internal oscillator with no axternai components.

TEST is an input, used only in testing the 3870. For normal circuit functionality this pin is ieft unconnected or may be grounded.

V_{CC} is tha power supply input (+5 ± 10%).



with Mask- Programmable Read-Only Memory (ROM)

MC141000/1200 Family

Somewhet less sophisticated then the MC3870, this *4-bit* single-chip microcomputer, nevertheless, is more than adequete for a wide range of epplicetions . . . end, it offers some unique edventeges. It features CMOS circuitry, providing the lowest possible power consumption, and making it suitable for battery powered, battery back-up, or conventionel 5 V operation.

Forty-three basic instructions hendle I/O, constant data from ROM, bit control, internal data transfer, arithmetic processing, logic comparison, conditionel and nonconditionel branching, and subroutines. A 1024 x 8-bit ROM and a 64 x 4-bit RAM handle the on-chip memory requirements.

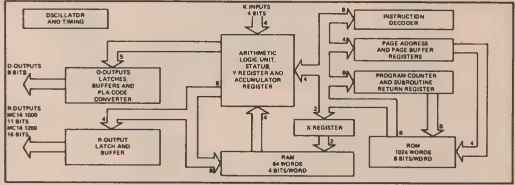
The MC141000/1200 Family is source-progrem compatible, pin-out compatible and architecturelly similar to the well-known PMOS TMS1000 Family, but with the following additional features:

- Power Consumption — only 2.5 mW at 5 V
only 500 μW at 3 V
- Fully Static Operation
- TTL-Competible — Drives One TTL Load or Four LSTTL Loads
- Clock Frequency to 700 kHz at VDD = 5 V
- 18 "R" Outputs (MC141200)

Applications

- Appliance Controllers
- Calculators
- Toys
- Radio Controllers
- Communications Controllers
- Data Tarminals
- Cash Registers
- Heating/Air-Conditioning Controllers
- Ramota Sensing System
- Printing Controllers
- Security Systems
- Powar Systems Control
- Automotive Control

Tha abova applications of the MC141000 Family demonstrate its wide potential. Motorola will accept customer programs or will contract complete program development, given the specifications for the application. Customer hardware and softwara support is alraady availabla for developing programs and dabugging systams This consists of one board and a software package using the M6800 EXORciser. Contact your local sales office for status and availability of support equipmant.


DSCILLATDR AND TIMING
K INPUTS 4 BITS
4
INSTRUCTIDN DECOOER
8
PAGE ADORESS AND PAGE BUFFER REGISTERS
4
PROGRAM COUNTER AND SUBROUTINE RETURN REGISTER
8
5
D-OUTPUTS 8 BITS
O-OUTPUTS LATCHES BUFFERS AND PLA CODE CONVERTER
ARITHMETIC LOGIC UNIT, STATUS, Y REGISTER AND ACCUMULATOR REGISTER
8
4
R DUTPUTS MC14 1000 11 BITS MC14 1200 16 BITS
4
R-OUTPUT LATCH AND BUFFER
2
X REGISTER
4
2
8
8
4
ROM 1024 WORDS 8 BITS/WDRD
RAM 64 WORDS 4 BITS/WORD
8

Single-chip microcomputers . . .

looking ahead- the MC6801

The first M6800 Microcomputer on a single chip

With expected availability before the end of '78, the MC6801 single-chip microcomputer merits serious attention for the next generation of equipment now being designed

Why?

Well, for starters, here are a few of the more important reasons.

It is characterized with all the circuitry basic to the M6800 MPU .

plus

The on-chip clock oscillator and 128 x 8-bit RAM of the MC6802 . .

plus

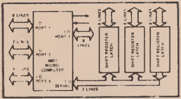
A 2K x 8-bit ROM

A 16-bit timer . .

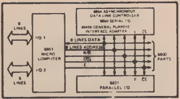
A vast expansion of I/O capacity.

There's not much left to chance in the architecture of the MC6801 There are 31 programmable parallel I/O lines for managing external peripherals, and a serial I/O port for controlling communications equipment. A powerful interrupt capability, with 8 interrupt levels, cuts design costs and boosts performance.

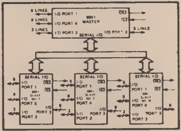
And, just in case the built-in capacity of the MC6801 is insufficient, for a specific application, it retains complete compatibility with the rest of the M6800 line. This means memory expandability up to 65K bytes and a wide variety of other functional options that makes an MC6801 system one of the most powerful in the industry.



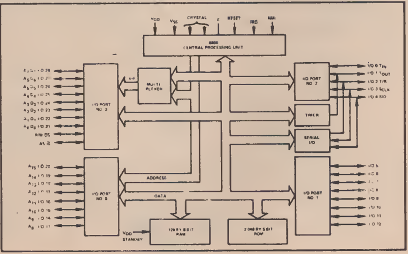
Low-cost Data Communications with simple serial I/O peripherals, such as shift registers is facilitated by an on-chip Serial I/O port



Expanding parallel I/O capacity is simplified by use of I/O ports to expand the chips data bus and address bus to external M6800-oriented peripherals



Using the Serial I/O port, a number of MC6801s can be arranged in a master-slave multiprocessor setup



Profile of an advanced single-chip microcomputer—The Motorola MC6801

Where high speed counts . . .

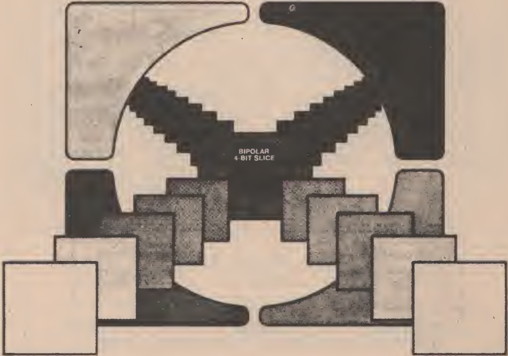
bipolar 4 bit-slice processors

Long before MOS technology became the unofficial standard for microcomputers, computers were being built with bipolar building blocks . . . and they still are. While suffering in comparison with MOS circuits in terms of processing simplicity and, therefore, cost (for LSI configurations), they have an overriding advantage in terms of speed. In many applications requiring real-time responses and complex calculations, bipolar processing represents the only alternative.

But even here, Motorola offers the system designer a choice—a choice between two bipolar circuit configurations: TTL (Schottky, of course) and ECL (Pages 20 and 21).

The bipolar approach to microcomputers differs considerably from the MOS approach. Rather than providing complete microcomputers, or even microprocessors, bipolar designers have evolved a "bit slice" concept. This consists of a series of LSI components representing various major functions of a general-purpose computer. These components are cascadable to form systems of any desired complexity. Thus, the basic 4-bit slice building blocks offered by the two Motorola bipolar families can be expanded into 8-bit, 16-bit and even 32-bit machines. Special circuit architecture makes these building blocks easily microprogrammable.

Unlike MOS MPUs, all of which are accompanied by proprietary software, the bipolar families can be designed to utilize software from any existing computer line.



Where high speed counts . . .

bipolar 4 bit slice processors

Speed and versatility are the key attributes of the bipolar 4-bit slice processor families when compared with the MOS components. Speed is the biproduct of bipolar processing; versatility results from the "slice" concept that permits virtually unlimited expansion of the computer system. Specifically, both families described here consist of 4-bit-wide components that are structured or "sliced" parallel to the data flow. This permits system expansion to larger word lengths simply by connecting several parts (of each type) in parallel.

The M2900 (LSTTL) Family
System Clock Frequency 8.3 to 9.5 MHz

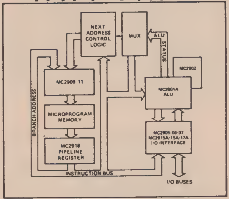
This family of TTL LSI components is microprogrammable for efficient emulation of almost any computing machine.

The heart of the system is the MC2901, a fully expandable 4-bit Arithmetic and Logic Unit (ALU). This device consists of a 16-word by 4-bit two-port RAM, a high-speed ALU, and the associated shifting, decoding and multiplexing circuitry. The ALU function has look-ahead or ripple carry, three-state outputs, and various status-flag outputs. The look-ahead carry function is performed with an MC2902 Look-Ahead Carry Generator in conjunction with the ALU.

The MC2909/2911 are four-bit wide address controllers intended for sequencing through a series of micro instructions contained in the microprogram memory. These controllers have a 4 x 4 stack that allows nesting of subroutines. The system speed can be improved by "pipelining" the contents of the microprogram into MC2918 four-bit registers. Also, the MC2918 register can be used as an address register, condition code register, or for various other register applications.

The I/O interface can be achieved with several different bus transceiver devices. The MC2905/06/07 have high-current sinking, open-collector bus outputs. The driver side has four D-type edge-triggered flip-flops and the receiver side has four D-type latches. The MC2915A 16A 17A are three-state bus output options. These bus transceivers can be used to transfer information from the ALU to the main memory or other bus applications.

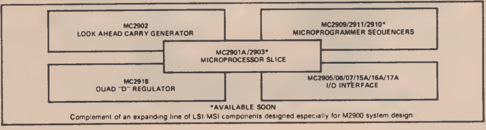
FASTER



Example of basic system implementation with dedicated M2900 components

COMPATIBLE TTL MEMORIES

	Size (Organization)	Device No.	Access Time (ns max)	P_D (mW typ/pkg)	Temperature (°C)
RAMs	1K Bits (1024 x 1)	MCM93415 MCM93425	35	550	0 to +70
ROMs	512 Bits (64 x 8)	MCM5003 MCM5303	75	500	0 to +70 -55 to +125
		MCM5004 MCM5304	75	600	0 to +70 -55 to +125
	4K Bits (512 x 8)	MCM7640 MCM7641	40 typ	500	0 to +70
	4K Bits (1024 x 4)	MCM7642 MCM7643			



MC2902
LOOK AHEAD CARRY GENERATOR

MC2909/2911/2910*
MICROPROGRAMMER SEQUENCERS

MC2901A/2903*
MICROPROCESSOR SLICE

MC2918
QUAD "D" REGULATOR

MC2905/06/07/15A/16A/17A
I/O INTERFACE

*AVAILABLE SOON

Complement of an expanding line of LSI/MSI components designed especially for M2900 system design

The M10800 (ECL) Family
System Clock Frequency 10 to 15 MHz

Offering the fastest cycle times of any available bit-slice processor family, the M10800 series of ECL 4-bit processor slices permits the design of high-speed computer systems.

The core of any M10800-based system is the Arithmetic and Logic Unit (ALU). It operates at system-clock frequencies of 10 to 15 MHz, which represent cycle times of 60 to 100 ns. System word size starts at the ALU width of 4 bits, but can be expanded to n × 4 bits by cascading ALU sections. To support the ALU, Motorola has developed several ECL circuits that take care of most of the housekeeping without restricting the processor design.

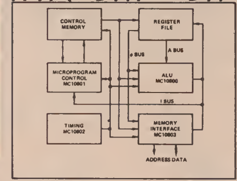
Intended to address the instructions stored in the microprogram memory, the MC10801 Microprogram Controller provides a 4-bit address that can be expanded to any size by cascading controllers. A memory interface unit, the MC10803, also has a cascadable 4-bit output bus, but it connects to the address bus of the main memory and supplies all the read and write addresses.

Acting as a register file, stack or I/O buffer, the MC10806 dual-port memory provides 32 words × 9 bits of temporary storage and can be accessed through either of its ports. For high-speed mathematical operations, the MC10808 Multibit Shifter can handle up to 16 bits and, under software control, can do left-shift, right-shift or rotate operations. Additional MC10808s can be cascaded for larger word lengths.

Other support circuits include the MC10802 Timing Generator and Clock Controller, the MC10804 and MC10805 Bidirectional Bus Translators (ECL-to-TTL and vice-versa) and all of the MECL 10,000 series of logic circuits.

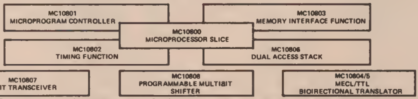
MECL is a trademark of Motorola Inc.

FASTEST



One example of high performance system implementation using basic M10800 Building Blocks.

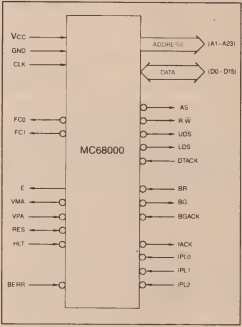
COMPATIBLE ECL MEMORIES					
	Size (Organization)	Device No.	Access Time (ns max)	PD (mW typ/pkg)	Temperature (°C)
RAMs	1K Bits (1024 x 1)	MCM10146	29	500	0 to +75
	256 Bits (256 x 1)	MCM10144	26	420	
		MCM10152	15	500	
	128 Bits (128 x 1)	MCM10147	12	420	-30 to +85
	64 Bits (16 x 4)	MCM10145	15	625	0 to +75
	16 Bits (8 x 2)	MCM10143	14.5	810	-30 to +85
ROMs	256 Bits (32 x 8)	MCM10139A	10 typ	500	0 to +70
	1K Bits (256 x 4)	MCM10149	25	540	-30 to +85



Present complement of an expanding line of LSI/MSI components designed especially for M10800 system design.

INTRODUCING THE MC68000 . . . MOTOROLA'S ADVANCED COMPUTER SYSTEM ON SILICON

The MC68,000 microprocessor is housed in a 64-pin package that allows the use of separate (non-multiplexed) address and data buses. This large package provides optimum flexibility while at the same time maximizing bus through-put.



PIN IDENTIFICATION & DEFINITIONS

A1–A23	Address Leads	23-bit address bus; capable of addressing 16,777,216 bytes in conjunction with UDS and LDS
D0–D15	Data Leads	16-bit data bus, transfers 8 or 16 bits of information
AS	Address Strobe	Indicates valid address & provides a bus lock for indivisible operations
R/W	Read/Write	Defines bus operation as Read or Write and controls external bus buffers
UDS, LDS	Data Strobes	Identifies the byte(s) to be operated on according to R/W and AS
DTACK	Data Transfer Acknowledge	Allows the bus cycle to synchronize with slow devices or memories.
BR	Bus Request	Input to the Processor from o device requesting the bus.
BG	Bus Grant	Output from the processor granting bus arbitration.
BGACK	Bus Grant Acknowledge	Confirmation signal from BG indicating a valid selection from the arbitration process
IACK	Interrupt Acknowledge	Identifies that the bus is performing an interrupt service cycle.
IPL0, IPL1, IPL2	Interrupt Priority Level	Provides the priority level of the interrupting function to the processor
FC0, FC1	**Function Code**	**Provides external devices with information about the** current bus cycle.
CLK	Clock	Master TTL input clock to the processor.
RES	Reset	Provides reset (initiolization) signal to the processor ond peripheral devices
HLT	Halt	Stops the processor and allows single stepping.
BERR	Bus Error	Provides termination of a bus cycle if no response or on invalid response is received
E	Enable	Enable clock for M6800 systems.
VPA	Valid Peripheral Address	Identifies addressed area as a 6800 compatible area
VMA	Volid Memory Address	Indicates to 6800 family devices that a valid address is on the bus.
V_{cc}	+5 Volts	
GND	Ground (2 pins)	

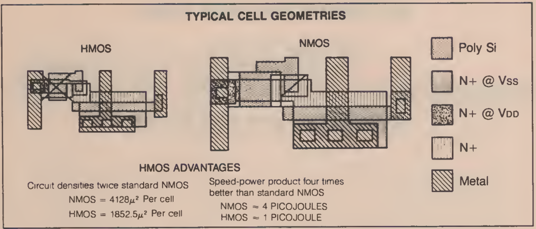

TYPICAL CELL GEOMETRIES
HMOS
NMOS
Poly Si
N+ @ Vss
N+ @ VDD
N+
Metal
HMOS ADVANTAGES
Circuit densities twice standard NMOS
NMOS = $4128\mu^2$ Per cell
HMOS = $1852.5\mu^2$ Per cell
Speed-power product four times better than standard NMOS
NMOS ≈ 4 PICOJOULES
HMOS ≈ 1 PICOJOULE

Figure 1: Comparison of HMOS and NMOS Technologies
HMOS Technology used for the MC68000 results in significant improvements to Circuit Densities and Speed-Power Products

Advances in semiconductor technology have provided the capability to put on a single silicon chip, a microprocessor at least an order of magnitude higher in performance and circuit complexity than has been previously available. The MC68000 is the first of a family of such VLSI microprocessors from Motorola. It combines state-of-the-art technology and advanced circuit design techniques with computer sciences to achieve an architecturally advanced 16-bit microprocessor containing over 68000 active devices on a silicon chip. This high density of active elements coupled with an order of magnitude increase in performance over the original MC6800 is the direct result of significant advances in semiconductor technology. Advances such as dry PLASMA etching, projection printing, and HMOS (High density short channel MOS) circuit design techniques (Figure 1) have provided a sound technological base that has allowed Motorola's system engineers, computer scientists and marketing engineers a large degree of innovative freedom. The goals of applying this innovative freedom to microprocessors are to make the microprocessor easy to use, more reliable and more flexible for applications, while maximizing performance.

The resources available to the MC68000 user consist of the following:

- 32-bit data and address registers
- 16 mega-byte direct addressing range
- 61 powerful instruction types
- operations on six main data types
- memory mapped I/O
- 14 addressing modes

Particular emphasis has been given to the architecture to make it orthogonal (regular) with respect to the registers, instructions (including all addressing modes), and data types. Orthogonality makes the architecture easy to learn and program, and, in the process, reduces both the time required to write programs and the space required to store programs. The net result is a great reduction in the cost and risk of developing software.

High systems throughput (up to an aggregate of two million instruction and data word transfers per second) is achieved even with readily available standard product memories with comparatively slow access times. The design flexibility of the data bus allows the mixing of slow and fast memories or peripherals with the processor, automatically optimizing the transfer rate on every access to keep the system operating at peak efficiency.

The hardware design of the CPU was heavily influenced by advances made in software technology. High level language compilers as well as code produced from high level languages must run efficiently on the new generation 16-bit and 32-bit microprocessors. The MC68000 supports high level languages with its consistent architecture, multiple registers and stacks, large addressing range and high level language oriented instructions (LINK, UNLINK, CHK, etc.). Also, operating systems for controlling the software operating environment of the MC68000 MPU are supported by privileged instructions, memory management, a powerful vectored multi-level interrupt and trap structure, and specific instructions (EXG, LDM, STM, TRAP, etc.).

The processor also provides both hardware and software interlocks for multiprocessor systems. The CPU chip contains bus arbitration logic for a shared bus and shared memory environment (shared with other MC68000 processors, DMA devices, etc.). Multiprocessor systems are also supported with software instructions (TEST and SET, TEST and RESET, etc.). The MC68000 offers the maximum flexibility for microprocessor based multiprocessor systems.

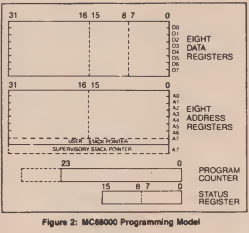


Figure 2: MC68000 Programming Model

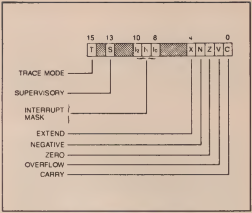


Figure 3: MC68000 Status Register

THE MC68000 CPU

Advanced architecture processors must not only offer efficient solutions to large complex problems but must be able to handle the small, simple problems with proportional efficiency. The CPU has been designed to offer the maximum in performance and versatility to solve simple and complex problems efficiently.

The MC68000 offers sixteen 32-bit registers in addition to the 24-bit program counter and 16-bit status register (Figure 2). The first eight registers (D0–D7) are used as data registers for byte (8-bit), word (16-bit) and long word (32-bit) operations. The second set of eight registers (A0–A7) may be used as software Stack Pointers and Base Address Registers. In addition, the second set of eight registers may be used for word and long word data operations. All of the sixteen registers may be used as Index Registers.

The 24-bit Program Counter provides a memory addressing range of more than 16 mega-bytes (actually 16,777, 216 bytes). This large range of addressing capability, coupled with a Memory Management Unit, allows large, modular programs to be developed and operated without resorting to cumbersome and time consuming software bookkeeping and paging techniques.

The Status Register (Figure 3) contains the Interrupt Level Mask (8 levels available) as well as the Condition Code; Overflow (V), Zero (Z), Negative (N), Carry (C), and Extend (X). Additional status bits indicate that the processor is in a TRACE (T) mode or in a SUPERVISORY (S) state . Ample space remains in the Status Register for future extensions of the M68000 family.

Six basic data types are supported. These data types are:

- Bits
- BCD digits
- ASCII characters
- Bytes (8-bits)
- Words (16-bits)
- Long words (32-bits)

In addition operations on other data types such as memory addresses, status word data, etc. are provided for in the instruction set.

DEFINITIONS

EA = Effective Address
Ax = Address Register
Dx = Data Register
Rx = Address or Data Register used as Index Register
SR = Status Register
PC = Program Counter
D_8 = Eight-Bit Offset
D_{16} = Sixteen-Bit Offset
N = 1 for Byte 2 for Word and 4 for Long Word
( ) = Contents of
← = Replaces

TABLE I: MC68000 DATA ADDRESSING MODES

REGISTER DIRECT ADDRESSING	
Data Register Direct	EA = D_x
Address Register Direct	EA = A_x
Status Register Direct	EA = SR
ABSOLUTE DATA ADDRESSING	
A. Absolute Short	EA = (Next Word)
B. Absolute Long	EA = (Next two Words)
PROGRAM COUNTER RELATIVE ADDRESSING	
Relative with Offset	EA = (PC) + D_{16}
Relative with Index & Offset	EA = (PC) + (Rx) + D_8

REGISTER INDIRECT ADDRESSING	
Register Indirect	EA = (Ax)
Post-increment Register Indirect	EA = (Ax), Ax ← Ax + N
Pre-decrement Register Indirect	Ax ← Ax − N, EA = (Ax)
Register Indirect with Offset	EA = (Ax) + D_{16}
Indexed Register Indirect with Offset	EA = (Ax) + (Rx) + D_8
IMMEDIATE DATA ADDRESSING	
Immediate	DATA = Next Word(s)
Quick Immediate	INHERENT DATA

The 14 flexible addressing modes, shown in Table I, include five basic types:

- Register Direct • Immediate
- Register Indirect
- Absolute • Program Counter Relative

Included in the addressing modes is the capability to do Post-incrementing, Pre-decrementing, Offsetting and Indexing.

THE INSTRUCTION SET

The MC68000 instruction set is rich and full as evidenced by the 61 distinct types shown in Table II. Special emphasis during the design has been given to the instruction set's support of structured high level languages that facilitate ease of programming. Each instruction, with few exceptions, operates on bytes, words, and long words and most instructions can use any of the 14 addressing modes. Combining instruction types, data types, and addressing modes, over 1000 useful instructions are provided. These instructions include signed and unsigned multiply and divide, "quick" arithmetic operations, BCD arithmetic and extended operations (through traps). The processor offers the most comprehensive and flexible instruction set of any microprocessor of any class, available today. Additionally, it's highly orthogonal, proprietary microcoded structure provides a sound flexible base for the future.

REDUCED SOFTWARE COST AND RISK

Advances in VLSI semiconductor technology have resulted in a significant reduction in the cost of computer hardware in recent years. The MC68000 microprocessor, for example, provides in a single integrated circuit package computing power that just a decade ago would have been three or four orders of magnitude more expensive. Software costs during this same period of time have, as a percentage of total system cost, increased significantly. This has been due primarily to inflation and the labor intensive nature of programming. Without significant architectural advances in computers, this trend can do nothing but continue. One of Motorola's major goals in developing this new microprocessor has been to reduce the costs of software. Many innovative features have been incorporated to make programming easier, faster and more reliable.

An Orthogonal 16-BIT MPU — The highly orthogonal or regular structure of the MC68000 microprocessor greatly simplifies the effort required to write programs in Assembly Language as well as in High Level Languages. Operations on integer data in registers and memory are independent of the data itself. Separate special instructions that operate on byte (8-bit), word (16-bit) and long-word (32-bit) integers are not necessary. The programmer merely has to remember one mnemonic for each type of operation and then specify data size, source addressing mode and destination addressing mode. This has helped keep the total number of instruction mnemonics for the M68000 to an easily remembered, yet complete, 61 types, eleven fewer than on Motorola's MC6800.

The dual operand nature of many of the instructions significantly increases the flexibility and power of this new Motorola microprocessor. Consistency again is maintained since *all* data registers and memory locations may be either a source or destination for most operations on integer data.

TABLE II: MC68000 INSTRUCTION SET SUMMARY

MNEMONIC	DESCRIPTION
ABCD	Add Decimal with Extend
ADD	Add
ADDX	Add with Extend
AND	Logical And
ASL	Arithmetic Shift Left
ASR	Arithmetic Shift Right
BCC	Branch Conditionally
BCHG	Bit Test and Change
BCLR	Bit Test and Clear
BRA	Branch Always
BSET	Bit Test and Set
BSR	Branch to Subroutine
BTST	Bit Test
CHK	Check Register Against Bounds
CLR	Clear Operand
CMP	Arithmetic Compare
DCNT	Decrement and Branch Non-Zero
DIVS	Signed Divide
DIVU	Unsigned Divide
EOR	Exclusive Or
EXG	Exchange Registers
EXT	Sign Extend
JMP	Jump
JSR	Jump to Subroutine
LDM	Load Multiple Registers
LDQ	Load Register Quick
LEA	Load Effective Address
LINK	Link Stack
LSL	Logical Shift Left
LSR	Logical Shift Right
MOVE	Move
MULS	Signed Multiply
MULU	Unsigned Multiply
NBCD	Negate Decimal with Extend
NEG	Two's Complement
NEGX	Two's Complement with Extend
NOP	No Operation
NOT	One's Complement
OR	Logical Or
PACK	Pack ASCII to BCD
PEA	Push Effective Address
RESET	Reset External Devices
ROTL	Rotate Left without Extend
ROTR	Rotate Right without Extend
ROTXL	Rotate Left with Extend
ROTXR	Rotate Right with Extend
RTR	Return and Restore
RTS	Return from Subroutine
SBCD	Subtract Decimal with Extend
SCC	Set Conditional
STM	Store Multiple Registers
STOP	Stop
SUB	Subtract
SUBX	Subtract with Extend
SWAP	Swap Data Register Halves
TAS	Test and Set Operand
TRAP	Trap
TRAPV	Trap on Overflow
TST	Test
UNLK	Unlink Stack
UNPK	Unpack BCD to ASCII

The addressing modes have been kept simple without sacrificing efficiency. All fourteen addressing modes operate consistently and are independent of the instruction operation itself. Additionally, all address registers may be used for the Direct, Register Indirect and Indexed addressing modes. (Immediate, Program Counter Relative and Absolute addressing by definition do not use address registers). For increased flexibility, any data register — as well as any address register — may be used as an Index Register. Address register consistency is maintained for stacking operations since any of the eight address registers may be utilized as User Program Stack pointers with the Register Indirect Post-increment/Predecrement addressing modes. Register A7, however, is a special register that, in addition to its normal addressing capability, functions as the System Stack Pointer when stacking the Program Counter and Status Register for subroutine calls, traps and interrupts; while in the supervisory mode.

Structured Modular Programming — The art of programming microprocessors has evolved rapidly in the past few years. Numerous advanced techniques have been developed to allow easier, more consistent and reliable generation of software. In general, these techniques require that the programmer be more disciplined in observing a defined programming structure such as modular programming. Modular programming allows a required function or process to be broken down in short modules or subroutines that are concisely defined and easily programmed and tested. Such a technique is greatly simplified by the availability of advanced macro assemblers and block structured High Level Languages such as PASCAL. Such concepts are virtually useless, however, unless parameters are easily transferred between and within software modules that operate on a reentrant and recursive basis. (To be reentrant a routine must be usable by interrupt and non-interrupt driven programs without the loss of data. A recursive routine is one that may call or use itself). The MC68000 microprocessor provides the necessary architectural features to allow efficient reentrant modular programming. The "LINK" and "UNLINK" instructions reduce subroutine call overhead in two complementary instructions by allowing the manipulation of linked lists of data areas on the stack. The "STM" (Store Multiple Registers) and "LDM" (Load Multiple Registers) instructions also reduce subroutine call programming overhead. These allow the loading or storing, via an effective address, multiple registers that are specified by the programmer. Sixteen software trap vectors are provided with the "TRAP" instruction and are useful in operating system call routines or user generated "macro routines." Other instructions that support modern structured programming techniques are PEA (Push Effective Address), LEA (Load Effective Address), RTR (Return to Restore) as well as the normal JSR, BSR and RTS.

Of course, the powerful vectored priority interrupt structure of the microprocessor allows straightforward generation of reentrant modular Input/Output routines. Eight maskable levels of priority with 192 vector locations provide maximum flexibility for I/O control. (A total of 256 vector locations are available for interrupts, hardware traps, and software traps.)

Improved Software Testability — One of the major tasks the system programmer encounters when writing software for microcomputers is the detection and correction of errors, or "debugging." The time taken to "debug" software nearly always exceeds the time it takes to write the software. In practice, the old 20/80 rule often applies: "The last 20% of the job requires 80% of the effort." The microprocessor incorporates several features that reduce the chance for errors. These features, such as Orthogonality and the Structured Modular Programming capability, have already been discussed.

Of major importance to the systems programmer are features that have been incorporated specifically to detect the occurrence of programming errors or "bugs." Several hardware traps, provided to indicate abnormal internal conditions of the MC68000 processor, detect the following error conditions:

Word access with an odd address
Illegal instructions
Unimplemented instructions
Illegal addressing mode
Illegal Memory access (bus error)
Overflow on divide (divide by zero)
Overflow condition code (separate instruction TRAPV)

Additionally, the sixteen software TRAP instructions may be utilized by the programmer to provide applications oriented error detection or correction routines.

An additional error detection tool is the CHK (Check Register Against Bounds) instruction used for array bound checking by verifying that $0 \leq (REG) <$ LIMIT. A trap occurs if the register contents are negative or greater than the limit.

Finally, the MC68000 includes a facility that allows instruction-by-instruction tracing of a program being debugged. This TRACE MODE results in a trap being made to a tracing routine after each instruction execution. The TRACE MODE is available to the programmer when the microprocessor is in the SUPERVISORY state as well as the USER state, but may only be entered while in the supervisory state. The SUPERVISORY/USER states provide an additional degree of error protection for the microprocessor by providing memory protection of selected areas of memory when an external memory management device is used.

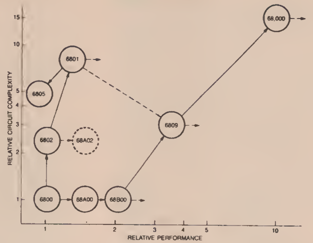


Figure 4:
Motorola's
Microprocessor Evolution

FUTURE FLEXIBILITY

Microprocessor VLSI circuit technology is advancing at an ever increasing rate. For example, the Motorola MC6800 — originally introduced in 1974 — has evolved into a number of more advanced products. This evolution has been along two paths: increased functionality, with the MC6802 and MC6801 microcomputers, and increased performance with the MC68A00, MC68B00 and MC6809 microprocessors. (Figure 4). The sound, well planned, architectural base provided by the original MC6800 made it possible to develop these improved products while taking full advantage of the major speed and density enhancements to NMOS VLSI. This was accomplished while maintaining an unprecedented degree of compatibility and consistency with the original MC6800 MPU.

Similarly, a major consideration in the development of the MC68000 microprocessor has been to provide a good, solid, but flexible, base for future extensibility. Several architectural concepts have been incorporated that will allow this advanced product to be enhanced as semiconductor technological advances are made. For example, the highly orthogonal structure of the CPU allows operations on 8-bit, 16-bit and 32-bit integers without the need for concatenation of registers or multiplexing of internal data buses. This regular structure of the CPU lends itself to a more consistent, reliable design that can be easily expanded.

The MC68000 incorporates a proprietary multilevel micro-programmed structure that allows significant versatility in the implementation of instructions. In fact, more than one-eighth of the instruction op-code map has been set aside specifically for implementation of future instructions. In the interim, **user implementation of instructions not currently in** the instruction set is possible through the use of the TRAP instruction, as well as the hardware trap structure.

MEMORY MANAGEMENT OF LARGE ADDRESSING SPACE

The ever-decreasing costs of semiconductor memories in combination with the use of high level languages and sophisticated disc operating systems allow Motorola's new generation of high performance microprocessors to be used in complex, memory intensive applications. In order to meet the needs of such applications, the MC68000 is capable of directly addressing more than 16 mega-bytes of memory. This large address space is directly accessed and managed very efficiently on a word or byte basis since operand size is specified by the instruction. The use of Upper Data Strobe (UDS) and Lower Data Strobe (LDS) signals allows easy access to high order bytes, low order bytes, or words.

Several additional useful features are provided that allow the programmer to efficiently manage memory usage. Powerful memory addressing modes such as Register Indirect, Indexed, Short and Long Absolute, and Program Counter Relative allow well-ordered access to specific memory locations. These addressing modes allow easy address calculations (Register Indirect and Indexed), direct access to memory location (Short and Long Absolute) and position independent or relocatable coding (Program Counter Relative). Of course, the Pre-decrement/Post-increment Register Indirect Addressing modes also allow efficient management of data in memory by permitting the programmer to generate as many as eight concurrent stacks or queues. Another feature that allows the programmer to manage the use of memory is the CHK (Check Register Against Bounds) Instruction. This instruction permits the software implementation of a basic memory protection/management structure.

Still another significant feature provided in the MC68000 microprocessor is the distinction between a USER and a SUPERVISOR mode. The SUPERVISOR mode permits certain protected operations within the processor system. Of particular interest is that an external Memory Management Controller may be used when the processor is in the USER mode to manage the large address space for the programmer. The controller's memory management operations are transparent to the programmer when in the USER mode and can be changed or updated only in the SUPERVISOR mode. The Memory Management Controller provides both management of a variable number of variable size segments (Memory Segmentation) and dynamic management of multi-task memory relocation and protection. The Memory Management Controller regulates access to storage segments that are dedicated to read only data, read/write data, program code and protected data/code.

REDUCED CODE DENSITY AND IMPROVED SPEED

With the advent of low cost, very high density VLSI RAMS and ROMS, it might incorrectly be assumed that the number of bytes of code needed to execute a given program is no longer important. Code density, however, is very critical, since microprocessor speed is highly dependent upon the number of executed instruction words. During the early development of Motorola's MC68000 microprocessor, extensive studies were made of the use of instructions and sequences of instructions in many microprocessor applications. These studies identified not only statically frequent instructions but also dynamically frequent instructions. (The dynamic frequency of instructions is a measure of how often an instruction is executed while static frequency is a measure of how often it occurs in a program listing or is encountered by an assembler). The major contributer to the increased efficiency, as a result of the studies, is the highly regular or orthogonal structure of the architecture. The consistency of the architecture, instruction set, and addressing modes significantly reduces the number of instructions needed to accomplish a given task. Additionally, many instructions have been included to specifically improve code density and speed. For example, single word Add and Subtract instructions using Quick Immediate addressing allow fast, small value arithmetic operations on data registers and memory. A Load Quick Immediate (LDQ) provides the ability to load a small (8-bit) signed word into any register in a single word operation. In order to improve the speed of loop operations, a single word instruction for Decrement Count by One and Branch if non-zero (DCNT) is included. Of course, the TRAP, Store Multiple Registers (STM), Load Multiple Registers (LDM), Link Stack (LINK), Unlink Stack (UNLK) and Check Limit (CHK) instructions significantly reduce code requirements for subroutines, operating system calls and stacking operations.

Other instructions that help reduce coding requirements and improve performance of arithmetic operations are Signed and Unsigned Multiply (MULS and MULU), Signed and Unsigned Divide (DIVS and DIVU), BCD Arithmetic (ABCD, SBCD, PACK and UNPK) as well as the standard binary integer operations. In order to improve the efficiency of moving or transferring data, a powerful MOVE data instruction has been incorporated that allows the transfer of bytes, words and long words and operates in all data addressing modes. Thus; register-to-register, register-to-memory, memory-to-register and memory-to-memory transfers are permitted.

In addition to the powerful instructions that provide a substantial improvement in processor through-put, numerous architectural features significantly reduce the execution times for all instructions. The separate (non-multiplexed) address and data buses, instruction pre-fetch pipeline and 32-bit internal registers are major contributors to the processor's unequaled performance. As an example of the performance capability of the MC68000 Table III and the accompanying graphs in figures 5 and 6 summarize the execution times for a number of common instructions. For comparison purposes, similar information is provided for Zilog's Z-8000 microprocessor. It is interesting to note that the MC68000 has significantly faster execution times.

TABLE III — EXECUTION TIMES FOR *MOVB R*, *SRC* INSTRUCTION FOR VARIOUS ADDRESSING MODES

Source Addressing	Motorola MC68000	Zilog Z-8000
Register	0 5us	0 75us
Indirect Register	1 0	1 75
Absolute Addressing (Direct)	1 5	2 25
Indexed Addressing	1 5	2 50
Immediate	1 0	1 00

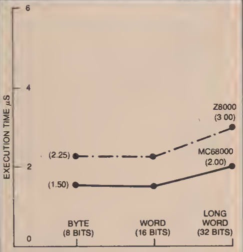


FIGURE 5: Execution Time for the *Add Data Element to a register from a short Absolute Address* **Instruction.**

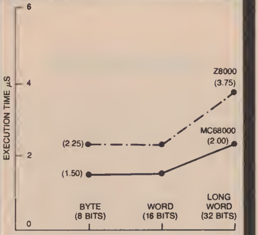


FIGURE 6: Execution Time for the *move a data element from memory to a register from short Absolute Address* **Instruction.**

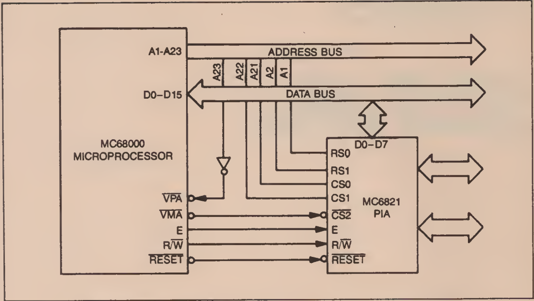


Figure 7: Example of MC68000 Interface Connections for MC6821 Peripheral Interface Adapter

SOFTWARE SUPPORT AND MC6800 COMPATIBILITY

The system designers and programmers using the MC68000 in an application have available a complete, compatible system of hardware and software. The microprocessor is supported by a full range of software development tools including disc operating systems, debug aids, assemblers, and high level languages. In addition, a translator will allow the present M6800 Family user to convert existing programs to run on the MC68000 with a minimum of programmer intervention.

The careful planning of this new microprocessor provides a superset of the MC6800 instruction set enhanced by the addition of more and larger registers, powerful orthogonal structure and many flexible addressing modes. This allows efficient translation of existing MC6800 programs, which can then be further optimized by taking full advantage of the versatile and powerful features of the MC68000.

This careful planning of similarities between the MC68000 and the MC6800 does not stop at software compatibility (by translation) but also extends to peripheral controller interfacing. Motorola's extensive line of intelligent M6800 family peripherals (including the MC6854 Advanced Data Link Controller and the MC68488 General Purpose Interface Adapter) can be directly and easily interfaced to the MC68000. Three signal lines; Enable (E), Valid Memory Address (VMA), and Valid Peripheral Address (VPA) are provided to simplify the interface to Motorola's standard MC6800 peripherals as shown in Figure 7. Interface to the new MC6801E (Single Chip Programmable Controller) is also possible, allowing user implementation of specialized input/output functions. In addition, the MC68000 is supported by unique peripheral controllers expected of an advanced architecture microprocessor, including a DMA Controller and a Memory Management Unit.

The MC68000 is not just a component. By a unique blend of VLSI design, software engineering and careful planning, the MC68000 it is Motorola's Advanced Computer System on Silicon.

MOTOROLA

MC14500B

INDUSTRIAL CONTROL UNIT

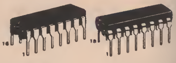
The MC14500B Industrial Control Unit (ICU) is a single bit CMOS processor. The ICU is designed for use in systems requiring decisions based on successive single bit information. An external ROM stores the control program. With a program counter (and output latches and input multiplexers, if required) the ICU in a system forms a stored program controller that replaces combinatorial logic. Applications include relay logic processing, serial data manipulation and control. The ICU also may control an MPU or be controlled by an MPU.

- 16 Instructions
- DC to 1.0 MHz Operation at V_{DD} = 5 V
- On Chip Clock (Oscillator)
- Executes One Instruction per Clock Cycle
- 3 V to 18 V Operation
- Noise Immunity Typically 45% of V_{DD}
- Quiescent Current 5.0 μAdc Typical at V_{DD} = 5 V
- Capable of Driving One Low-Power Schottky Load or Two Low-Power TTL Loads over Full Temperature Range

CMOS LSI

(LOW-POWER COMPLEMENTARY MOS)

INDUSTRIAL CONTROL UNIT



L SUFFIX
CERAMIC PACKAGE
CASE 620

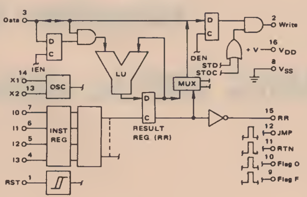
P SUFFIX
PLASTIC PACKAGE
CASE 648

ORDERING INFORMATION

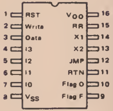
MC14XXXB __ __ Suffix Oanotas
- L Ceramic Package
- P Plastic Package
- A Extended Oparating Tamperatura Ranga
- C Limited Operating Tamperatura Range

Detailed operation and applications are given in the "MC14500B Industrial Control Unit" handbook.

BLOCK DIAGRAM



PIN ASSIGNMENT



MOTOROLA

THE MC14500B INDUSTRIAL CONTROL UNIT

A 1-Bit CMOS Microprocessor

Many of the tasks performed with today's classical general-purpose multi-bit microprocessors, or with hardwired logic, are really one-bit tasks. For these, multi-bit processors (which must be programmed to look like a one-bit processor) represent a needless "overkill", while with hardwired logic, the versatility needed to adapt a circuit to a variety of system requirements is not readily available. To perform such tasks in the most efficient and economical manner, Motorola has developed the MC14500B *Industrial Control Unit.*

Applications

The MC14500B, Industrial Control Unit is a one-bit processor that operates over a full voltage and temperature range of CMOS JEDEC B-Series parts. It has as its peripheral components the whole CMOS family of over 100 parts. This allows tailoring a system to an application, and permits a judicious mix of customized hardware and software to be achieved. As a one-bit processor it can be applied in a multitude of systems such as:

- PLC (Low Cost) Programmable Logic Controllers
- Machine Controls
- Numerical Controllers
- Industrial Controls
- Traffic Controllers
- Copier Controllers
- Automatic Test Systems
- Telephone Dialing Systems
- Serial-Bit Stream Communications Systems
- Remote-Bit Stream Controllers
- Serial Data Processors
- Commercial Product Controllers
- Automotive Systems
- Microprogram Control Sequencer
- Peripheral Controllers, Printers, Keyboards, Discs, etc.
- MPU Companion for Unloading Overtaxed μPs

There are functions, however, for which one-bit machines are poorly suited. These functions normally include complex calculations or parallel-word data processing. On the other hand, when the task is decision-and-command oriented, a one-bit machine is an excellent choice. The tasks that are mixed between decisions and calculations will be decided upon by economics, the designer's familiarity with alternatives, and how comfortable the designer is with the alternatives. Under some circumstances, a combination of an MC6800 MPU and an MC14500B ICU may be the best solution.

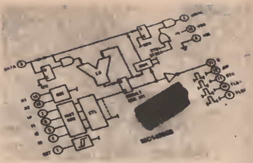


FIGURE 1 – BLOCK DIAGRAM

A 100+-page Handbook is available, detailing the operation and applications of the ICU

The Motorola MC14500B is a single chip, one-bit static CMOS processor optimized for decision-oriented tasks. Many of these decision-oriented tasks were well performed by relays, but with the MC14500B, electronic versatility can now be achieved in cost-effective systems. It is housed in 16-pin packages and features 16 four-bit instructions. The instructions perform logical operations on data appearing on a one-bit bidirectional data line and data in a one-bit accumulating Result Register within the ICU. All operations are performed at the bit level. The ICU has inherent CMOS qualities of high noise-immunity, microwatt power dissipation, the ability to use low-cost, low-current power supplies, 3 to 18 volt operating range, and the ease of battery backup and battery operation.

The ICU is timed by a single phase clock signal, generated by an internal oscillator that uses one external resistor.

Alternatively, it may be driven by an external source. In the external timing mode, the Clk signal is driven into pin 13 of the chip and the Clk signal is available as an output on pin 14 for synchronization with other systems.

In either case, the clock signal is available for synchronization with other systems. Each of the ICU's instructions execute in a single clock period. The clock frequency may be varied over a wide range. At a clock frequency of 1.0 MHz, more than 8300 instructions may be executed in a 60-Hz half-cycle.

The MC14500B instruction set consists of 16 instructions, as shown on page 4, each of which executes in one Clk period. The operating frequency range is from dc (single stepping) to a typical frequency of 1 MHz at 5 volts. The circuit's ability to execute instructions in one clock period, combined with its speed capabilities, allow it to outperform many of the more complex microprocessors for decision-oriented tasks.

Circuit Operation

The MC14500B processor operates synchronously with a single-phase clock. The clock divides the machine cycle into two periods. The first period (Clk High) is the "fetch" period, during which external memory (ROM) supplies the processor with an instruction. The instruction will be latched into the Instruction Register (IR) on the falling edge of the clock signal. The second period (Clk Low) is the "execu-

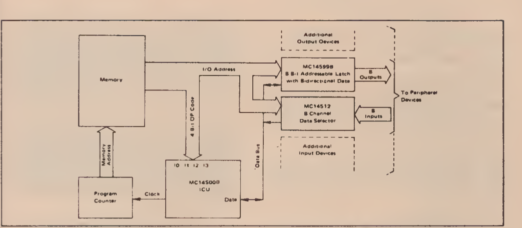


FIGURE 2 – OUTLINE OF A TYPICAL ORGANIZATION FOR A MC14500B BASEO SYSTEM

tion" period, during which the processor performs the command latched in its IR. During the execution period of an *output instruction*, the MC14500B puts the data in the Result Register (RR) onto the data line and raises the Write control line. The data must then be latched by the output circuitry at the end of the execution period of a machine cycle; i.e. on the rising edge of the (Clk) signal.

During the execution period of *input or logical commands*, the data present on the data line at the time the Clk signal is low is accepted by the ICU; the logic operation is then performed and the result is clocked into the Result Register on the rising edge of the clock (Clk) signal. This "split cycle" architecture makes it possible to "interleave" instructions and operands addressed in memory, the advantage being that the address lines of the I/O ports can share the ICU's instruction lines, forming a common bus.

The System

Figure 2 shows a block diagram of a minimal MC14500B system configured with standard B series logic parts. The blocks of the system are:

The MC14500B — which serves as the central controller of the system;

The ROM — which holds the instructions and the operand addresses;

The program counter — used to step the machine through the sequence of instructions stored in memory;

The input selector (demultiplexer) — used to route the addressed input to the MC14500B's one-bit bidirectional data bus;

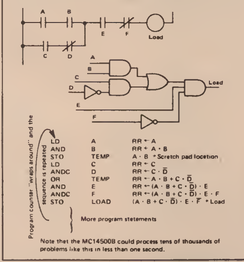
The output latches — which receive data placed on the 1-bit bidirectional bus by the MC14500B when an output instruction is executed.

The system can easily be expanded in terms of I/O, so long as the memory is sufficiently wide to address the I/O structure.

A CMOS PLC

MC14500B Industrial Control Unit is essentially the monolithic embodiment of the Programmable Logic Controllers (PLC) central architecture. Its Logic Unit (LU) is capable of doing a number of different Boolean functions under the control of instructions latched into the Instruction Register. The LU has two inputs: Data from the "Outside World" and the output of a Result Register.

The output of the LU is latched into the Result Register, where the new result will serve as one of the LU inputs on the next instruction. In general:



```
LD      A       RR ← A
AND     B       RR ← A · B
STO     TEMP    A · B  → Scretch pad locetion
LD      C       RR ← C
ANDC    D       RR ← C · D̄
OR      TEMP    RR ← A · B + C · D̄
AND     E       RR ← (A · B + C · D̄) · E
ANDC    F       RR ← (A · B + C · D̄) · E · F
STO     LOAD    (A · B + C · D̄) · E · F̄  → Load
        }
        }   More program stetements
        }
```

Note thet the MC14500B could process tens of thousands of problems like this in less than one second.

FIGURE 3 – COMBINATIONAL LOGIC PROBLEM

$$RR_{new} = f\ (\text{data}, RR_{old}).$$

After calculating a Boolean equation, the state of the Result Register or its complement is stored in an output latch to activate or deactivate load devices.

Looping Control Structure

A typical problem example is illustrated in Figure 3. The solution shown illustrates a looping control structure. Here, it is seen that the MC14500B continuously samples its inputs and, based on that information, transfers a logic signal to activate or deactivate the load device.

The ICU does not alter the sequential operation of the program counter. It sequentially fetches instructions from memory and when the program counter reaches its highest value it "wraps around," and the entire program is repeated, effecting what is known as a "looping control structure."

Time Invariant Software

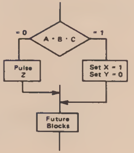
What is important beyond the concept of a "looping control structure" is the fact that the ICU can effect a "conditional jump" without parallel loading the program counter with the Jump address. The concept is that code requiring conditional branching can be written even though all code is fetched from memory sequentially. To accomplish this an output mask is used in the MC14500B. The Output Enabling Register — OEN, may be set or reset with the result of an evaluated logic equation. If the mask is reset, the Write strobe is inhibited and no change can be made in the state of the output devices. The processor can therefore use its logic power to turn off or on whole blocks of instructions. This leads to a looping control structure in which the same sequence of commands are encountered with certain blocks of code being selectively enabled or disabled.

Figure 4 shows a flow chart and program solution of a typical problem. In this example, the ICU uses its OEN mask to effect a "pseudo" branch. It first resolves the question asked in the decisions block (statements 1, 2, and 3), then stores the result (1 = true, 0 = false) in a temporary storage location called (Flag). Next, the ICU loads its output mask with the result of the decision block (statement 5); if the OEN mask was set true, the following two output instructions (statement 6 and 7) would be "active" and the right branch of the decision block would be executed. If the mask was loaded with a zero, statements 6 and 8 would not change the state of the X and Y outputs effecting the Pseudo branch. If the mask was indeed loaded with a zero, this would mean the left branch of the decision block is to be executed. To do this the ICU loads the logical complement of Flag into the output mask (statements 8 and 9). Then if Flag was originally false (i.e. A • B • C = 0), the instructions following statement 9 will be active and statement 10 and 11 will create the pulse to be sent to output Z. Statements 12 and 13 force the output mask to the 1 state so that future blocks of code will not inadvertently be turned off.

One of the disadvantages of the conventional conditional jumps as a means of making decisions is that the execution time of the program varies with the state of the input signals. A looping program is inherently time invariant. This allows for synchronization with other systems and other advantages such as stable sync pulses for trouble shooting. In the real-world environment of training people and maintaining products, this can be worth a great deal.

Adding a Conventional Control Structure

In some control applications it may be advantageous to have a control structure like that of a conventional processor, rather than a looping control structure. This situation may occur when the system becomes large and timing is critical. Having the capability to call subroutines also helps to modularize the software. An MC14500B system can be easily modified to incorporate a jumping, conditional branching, and subroutine capability.



```
          1             LD     A      *RR ← A
          2             AND    B       RR ← A · B
          3             AND    C       RR ← A · B · C
          4             STO    Flag    Set Flag = 1 Il Right Branch Taken
     5    Right      ⎧  OEN    RR      Output Mask ← (A · B · C) = RR
          branch     ⎪
     6    block      ⎨  STO    X       1 → X
          of         ⎪
     7    code       ⎩  STOC   Y       0 → Y
     8                  LDC    Flag    If Flag was 0 the follow-      RR  ← Flag
                                       ing code will be active        OEN ← Flag
     9    Left       ⎧  OEN    RR
          branch     ⎪
    10    block      ⎨  STO  ⎧ Z       Creates Pulse
          of         ⎪       ⎨           ⎍
    11    code       ⎩  STOC ⎩ Z
    12                  ORC    RR      RR ← 1 = RR + R̄R̄   ⎫  Enable Future
    13                  OEN    RR      Output Mask ← 1    ⎭  blocks of code

          *RR = Result Register
```

FIGURE 4 – BLOCK ENABLING STRUCTURE

The MC14500B has three program control instructions which are intended for the purpose of adding conventional jumping, conditional branching, and sub-routine capabilities to an ICU system. These instructions condition the ICU to take the appropriate action and provide the necessary control signals to external logic circuits which actually perform the address modifications.

For More Information . . .

Comprehensive information on these and other topics is contained in the MC14500 ICU Applications Handbook. The book approaches each topic from basic fundamentals and builds on these toward more advanced levels. It is richly illustrated with many examples and includes system schematics and programming examples. Some topics covered are:

1. Basics Concepts
2. Basic Programming and Instruction Set
3. Hardware Systems
4. A Demonstration System
5. Signal conditioning and I/O Interface Circuits
6. Advanced Programming
7. A Comprehensive Example
8. Advanced Hardware Systems
9. Useful Routines

MC14500B OPERATIONS ANO COOES

Instruction Code			
Hex	Binery	Mnemonic	Action
0	0000	NOPO	**No Operation Zaro** No change in eny register The Fleg O output pin gets a one-period pulse beginning on the falling edge of X1 (Clk)
1	0001	LO	**Load** The Result Registar is loaded with the stete of the date signel
2	0010	LDC	**Load Complemented** The Result Register is loaded with the complement of the date signel.
3	0011	ANO	**AND** The Result Register is loaded with the logical ANO of the dete signal end the old Result Register stete.
4	0100	ANOC	**AND Complement** Tha Result Register is loaded with the logicel ANO of the old Result Register state end the complement of the dete signel
5	0101	OR	**OR** The Result Register is loaded with the logical OR of the dete signel end the old Result Register stete.
6	0110	ORC	**OR Complement** The Result Register is loaded with logical OR of the old Rasult Register state and tha complement of the deta signal.
7	0111	XNOR	**Exclusive NOR** The Result Register is loeded with a logical 1 if the old Result Register state and the date signal agree If the old result and the dete signal ere not alike, tha Result Register is loaded with e logicel 0.
8	1000	STO	**Stora** The Date pin is driven to tha state of the Result Register The Writa pin is driven high for e half period beginning with the fell of X1. The stete of the Result Register is not chenged
9	1001	STOC	**Store Complemented** Same es Stora, except the data signel is driven with the complement of the Result Register
A	1010	IEN	**Input Enable** The Input Enable Register is loadad with the state of the data signal
B	1011	OEN	**Output Enable** The Output Enable Register is loeded with the state of the data signal.
C	1100	JMP	**Jump** A one period pulse is generated et the JMP pin beginning with the falling edge of X1
O	1101	RTN	**Return** A one period pulse is genereted at the RTN pin beginning with the falling edge of X1, end the next instruction is ignored.
E	1110	SKZ	**Skip If Zaro** If the Result Register contains a logical 0 et the time of the instruction the next instruction is ignored.
F	1111	NOPF	**No Operetion F** No chenge in any register Flag F output pin gets a one period pulse beginning on the falling edge of X1 (Clk)

MOTOROLA

MC14 1000
MC14 1200

Product Preview

ONE CHIP MICROCOMPUTER

The MC14 1000 and the MC14 1200 are two members of the MC14 1000 family of CMOS 4-bit microcomputers. They incorporate ROM, RAM, ALU, control, and I/O in a single CMOS monolithic structure. The MC14 1000/2000 can be tailored to its application by internally programmed ROM and output PLA.

The MC14 1000 family is source program compatible, pin-out compatible, and architecturally similar to the PMOS TMS1000 family. These CMOS one-chip microcomputers offer the following additional features not available in the TMS1000:

- Low Power Consumption Suited for Battery-powered or Battery Back-up Systems
- Fully Static Operation
- TTL Compatible – Drives One TTL Load or Four LSTTL Loads
- Clock Frequency to 600 kHz at V_{DD} = 4.75 V
- Single Supply,,3 to 6 Volt Operation
- 16 "R" Outputs (MC14 1200)

FEATURES:

	MC141000	MC141200
Package Pin Count	28 Pins	40 Pins
Instruction Read Only Memory	1024 X B Bits (8,192 Bits)	
Data Random Access Memory	64 X 4 Bits (256 Bits)	
"R" Individually Addressed Output Latches	11	16
"O" Parallel Latched Data Outputs	8 Bits	
Maximum-Rated Voltage	6.5 V	
Working Registers	Static 2–4 Bits Each	
Instruction Set	See Table 1	
On-Chip Oscillator	Yes	
5 V Power Supply/Typical Dissipation	5 V/2.5 mW	
3 V Power Supply/Typical Dissipation	3 V/500 µW	

APPLICATIONS:

- Appliance Controllers
- Calculators
- Toys
- Radio Controllers
- Communications Controllers
- Data Terminals
- Cash Registers
- Heating/Air-Conditioning Controllers
- Remote Sensing System
- Printing Controllers
- Security Systems
- Power Systems Control
- Automotive Control

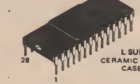
The above applications of the MC14 1000 family demonstrate its wide potential. Motorola will accept customer programs or will contract complete program development given the specifications for the application. Customer hardware and software support is available for developing programs and debugging systems. This consists of one board and a software package using the M6800 EXORciser. Contact your local sales office for details on the support equipment and software.

This is advance information and specifications are subject to change without notice.

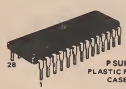
CMOS LSI

(LOW-POWER COMPLEMENTARY MOS)

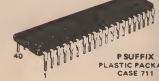
ONE CHIP MICROCOMPUTER



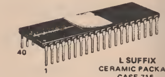
L SUFFIX
CERAMIC PACKAGE
CASE 719



P SUFFIX
PLASTIC PACKAGE
CASE 710



P SUFFIX
PLASTIC PACKAGE
CASE 711



L SUFFIX
CERAMIC PACKAGE
CASE 715

ORDERING INFORMATION

MC14XXXX — Suffix Denotes
- L Ceramic Package
- P Plastic Package

MAXIMUM RATINGS (Voltages referenced to V_{SS})

Rating	Symbol	Value	Unit
DC Supply Voltage	V_{DD}	-0.5 to +6.5	Vdc
Input Voltage, All Inputs	V_{in}	-0.5 to V_{DD} + 0.5	Vdc
DC Current Drain per Pin, All Inputs	I	10	mAdc
DC Current Drain, V_{DD} Pin	I	250	mAdc
DC Current Drain, V_{SS} Pin	I	20	mAdc
Dperating Temperature Range	T_A	-40 to +85	°C
Storage Temperature Range	T_{stg}	-65 to +150	°C
Total Power Dissipation MC14 1000 @ 25°C MC14 1200	P_D	400 600	mW

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \leq (V_{in}$ or $V_{out}) \leq V_{DD}$.

RECOMMENDED OPERATING CONDITIONS (V_{SS} = 0)

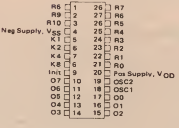
Parameter	Symbol	Value	Unit
DC Supply Voltage — High Speed Clock Full Range Operation	V_{DD}	+4.75 to +6.0 +3.0 to +6.0	Vdc
Clock Frequency — V_{DD} = 5.0 Vdc ± 5% V_{DD} = 3.0 Vdc Min.	f_{Clk}	DC to 600 DC to 200	kHz

ELECTRICAL CHARACTERISTICS (V_{DD} = +5.0 V, V_{SS} Gnd, T_A = 25°C unless otherwise noted)

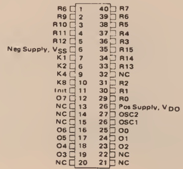
Characteristic	Symbol	Min	Typ	Max	Unit
Input Current — K Inputs (V_{in} = 5.0 V) (V_{in} = 0.0 V)	I_{in}	75 —	100 -0.00001	135 -0.3	µAdc
Dutput Drive — R and O Outputs (V_{OH} = 2.4 V) (V_{OL} = 0.4 V, T_A = 85°C, V_{DD} = 4.75 V)	I_{OH} I_{DL}	-20 1.6	— —	— —	mAdc
Average Supply Current (f_{Clk} = 500 kHz)	I_{DD}	—	—	1500	µAdc
Static Supply Current (V_{DD} = 6 V)	I_{DD}	—	60	300	µAdc
Dscillator Frequency (V_{DD} = 4.75 V)	f_{Clk}	No Limit	—	600	kHz
Internal Oscillator Frequency for R_{ext} = 30 kΩ	f_{Clk}	400	500	600	kHz
Input Capacitance — K Inputs	C_{in}	—	—	7.5	pF
Input Capacitance — Clock Input	C_{io}	—	—	30	pF

PIN ASSIGNMENTS

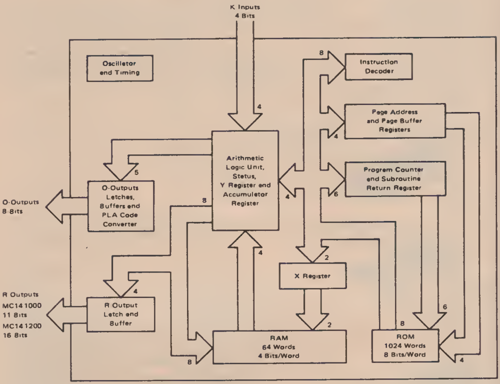
MC14 1000



MC14 1200



LOGIC BLOCK DIAGRAM


K Inputs
4 Bits
4
Oscillator and Timing
8
Instruction Decoder
Page Address and Page Buffer Registers
4
Arithmetic Logic Unit, Status, Y Register and Accumulator Register
5
O-Outputs Latches, Buffers and PLA Code Converter
O-Outputs
8 Bits
8
4
Program Counter and Subroutine Return Register
6
4
2
X Register
R Outputs
MC141000
11 Bits
MC141200
16 Bits
4
R Output Latch and Buffer
2
RAM
64 Words
4 Bits/Word
8
6
8
ROM
1024 Words
8 Bits/Word
4

The MC141000/1200 ROM program controls data input, storage, processing, and output. The processing of data occurs in the arithmetic logic unit (ALU). K input data enters the ALU and is stored in the 4-bit accumulator. The accumulator output accesses the output latches, the RAM storage cells, and the adder input. Data is stored in the 256-bit RAM, organized into 64 words, 4 bits per word. The 4-bit words are grouped into four 16-word files addressed by a 2-bit X register. A 4-bit Y register addresses one of the 16 words in a file by control signals from the ROM.

The 43 basic instructions handle I/O, constant data from the ROM, bit control, internal data transfer, arithmetic processsing, logic comparison, conditional and nonconditional branching and subroutines.

The designer has access for programming the following functions:

1) ROM – 1024 words of 8 bits
2) Program Logic Array for O outputs
3) Output circuits

PROGRAMMABLE ROM

The 1024 words of the 8-bit ROM are divided into 16 pages of instructions with 64 instructions on each page The program starts at the top of the sixteenth page A binary program counter sequentially addresses each ROM instruction on a page. One level of subroutine return address is stored. The page address register (4 bits) stores the current address for one of the 16 ROM pages. To change pages, a constant from the ROM loads into the page buffer register (4 bits), and upon a branch or call, the page buffer loads into the page address register.

RAM

The 64-word by 4-bit RAM comprises 4 files, each file containing 16 four-bit words. The RAM is addressed by the Y register and X register. The Y register selects one of the 16 words in a file and the X register selects one of four files. Any 4-bit word can be read or written. Any selected bit in the RAM can be set, reset, or tested.

INPUT

The 4 data inputs are designated K1, K2, K4, and K8. The R outputs can multiplex inputs for keyboard use. The R outputs can be used for handshake for control of the input from other devices.

The K inputs are static-protected CMOS inputs with pulldown of about 50 k ohms. Thus, an open input is equivalent to logic 0. The circuit is shown in Figure 1.

OUTPUT

The O-outputs comprise the output bus. The R-outputs are used as control lines to scan keyboards and displays, perform handshakes, and interface external logic. The 8 parallel O-outputs are decoded from the 5 bits in the O-output latches. This decoding is defined by the customer and is accomplished in the O-PLA. The 11 R-outputs of the MC14100 and the 16 R-outputs of the MC141200 are individually settable and resettable under program control.

The outputs may be programmed in one of two configurations, either open emitter or active sink NMOS as shown in Figures 2 and 3. The circuits of Figures 2 and 3 sources 20 mA @ V_O = 2.0 Volts and V_{DD} = 5.0 Volts. The circuit of Figure 3 sinks 1.6 mA over temperature to operate one TTL load or four LSTTL loads.

INTERNAL OR EXTERNAL CLOCK

The internal oscillator is controlled by the value of one resistor. This is an improvement over the PMOS TMS1000/1200. For external clocking, the clock signal is connected to Osc. 1. The oscillator circuit works with quartz crystals, ceramic resonators and L-C resonant circuits. Figure 4 shows the typical operation of the oscillator as a function of the programming resistor

FIGURE 1 – INPUT CIRCUIT WITH PULLDOWN AND STATIC PROTECTION

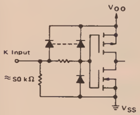


FIGURE 2 – OPEN EMITTER OUTPUT CIRCUIT

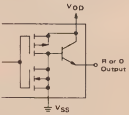
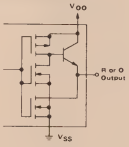


FIGURE 3 – ACTIVE NMOS OUTPUT CIRCUIT



CONNECTION FOR INTERNAL OSCILLATOR

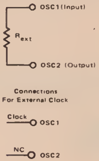


FIGURE 4 – TYPICAL OSCILLATOR FREQUENCY vs EXTERNAL RESISTANCE

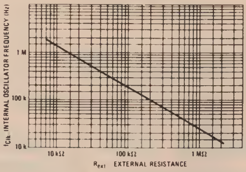


TABLE 1 – Standard Instruction Set

		Status Effects		
Function	Mnamonic	CARRY	COMPARED	Description
Ragister to	TAY			Transfer accumulator to Y register.
Register	TYA			Transfer Y register to accumulator.
	CLA			Clear accumulator.
Transfer	TAM			Transfer accumulator to mamory.
Registar to	TAMIY			Transfer accumulator to memory and increment Y register.
Memory	TAMZA			Transfer accumulator to mamory and zero accumulator.
Mamory to	TMY			Transfer memory to Y register
Ragistar	TMA			Transfer memory to accumulator.
	XMA			Exchange memory and accumulator.
Arithmetic	AMAAC	√		Add memory to accumulator, results to accumulator. If carry, one to status
	SAMAN	√		Subtract accumulator from memory, results to accumulator. If no borrow, one to status.
	IMAC	√		Increment memory and load into accumulator. If carry, ona to status.
	DMAN	√		Decrement memory and load into accumulator. If no borrow, one to status.
	IA			Incrament accumulator, no status effect.
	IYC	√		Increment Y register. If carry, ona to status.
	DAN	√		Decrement accumulator. If no borrow, one to status.
	DYN	√		Decrement Y register. If no borrow, one to status.
	ABAAC	√		Add B to accumulator, results to accumulator. If carry, one to status.
	A10AAC	√		Add 10 to accumulator, results to accumulator. If carry, one to status.
	A6AAC	√		Add 6 to accumulator, results to accumulator. If carry, one to status.
	CPAIZ	√		Complement accumulator and increment. If then zero, one to status
Arithmetic	ALEM	√		If accumulator less than or equal to memory, one to status.
Compare	ALEC	√		If accumulator less than or equal to a constant, one to status.
Logical	MNEZ		√	If memory not equal to zero, one to status.
Compare	YNEA		√	If Y register not equal to accumulator, one to status and status latch.
	YNEC		√	If Y register not equal to a constant, one to status.
Bits in	SBIT			Set memory bit.
Memory	RBIT			Reset memory bit.
	TBIT1		√	Test memory bit. If equal to one, one to status.
Constants	TCY			Transfer constant to Y register.
	TCMIY			Transfer constant to memory and increment Y.
Input	KNEZ		√	If K inputs not equal to zero, one to status.
	TKA			Transfer K inputs to accumulator.
Output	SETR			Set R output addressed by Y.
	RSTR			Reset R output addressed by Y.
	TDO			Transfer data from accumulator and status latch to O outputs.
	CLO			Clear O-output register
RAM 'X'	LDX			Load 'X' with a constant
Addressing	COMX			Complement 'X'
ROM	BR			Branch on status = one
Addressing	CALL			Call subroutine on status = one.
	RETN			Raturn from subroutine.
	LDP			Load page buffer with constant

NOTE: If tha bits COMPARED are not equal, or if there is a CARRY from the MSB of the adder, status will stay at one. Otherwise, status will go to a zaro for ona instruction cycla. Branch and Call can only execute successfully when status is á one The check marks (√) indicata the instructions that affect the status.

MOTOROLA

MC2901A

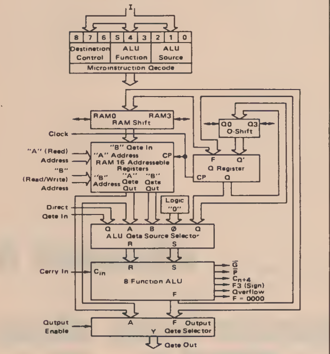
FOUR-BIT BIPOLAR MICROPROCESSOR SLICE

The four-bit bipolar microprocessor slice is designed, as a high-speed cascadable element intended for use in CPUs, peripheral controllers, programmable microprocessors, and numerous other applications. The microinstruction flexibility of the MC2901A will allow efficient emulation of almost any digital computing machine.

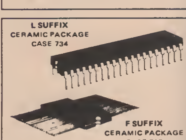
The device, as shown in the block diagram below, consists of a sixteen-word by four-bit, two-port RAM; a high-speed ALU; and, the associated shifting, decoding, and multiplexing circuitry. The 9-bit microinstruction word is organized into three groups of three bits each and selects the ALU source operands, the ALU function, and the ALU destination register. The microprocessor is cascadable with full look-ahead or with ripple carry, has three-state outputs, and provides various status flag outputs from the ALU. Advanced low-power Schottky processing is used to fabricate this 40-lead LSI chip.

- Plug-in Replacement for MC2901
- 20% to 30% Faster Than MC2901 in Most System Configurations
- Major Speed Improvements in D Input and Carry Paths
- I_{OL} Raised to 20 mA on Y Outputs — 30% More Drive Than MC2901
- I_{CC} Reduced to 190 mA at 125°C — 30% Less Than MC2901
- V_{IL} Raised to 0.8 V Over Full Military Range for Increased Noise Immunity

MICROPROCESSOR SLICE BLOCK DIAGRAM

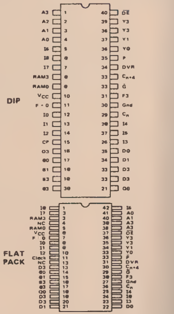


TTL
FOUR-BIT BIPOLAR MICROPROCESSOR SLICE



L SUFFIX
CERAMIC PACKAGE
CASE 734

F SUFFIX
CERAMIC PACKAGE
CASE 735



CAUTION

MC2901AFM pinout differs from MC2901FM on pins 4, 11, 12, and 13.

ORDERING INFORMATION

Package Type	Temperature Range	Order Number
Hermetic DIP	0°C to +70°C	MC2901ALC
Hermetic DIP	−55°C to +125°C	MC2901ALM
Hermetic Flat Pack	−55°C to +125°C	MC2901AFM

MOTOROLA

MC2902

HIGH-SPEED LOOK-AHEAD CARRY GENERATOR

The MC2902 is a high-speed, look-ahead carry generator which accepts up to four pairs of carry propagate and carry generate signals and a carry input and provides anticipated carries across four groups of binary ALUs. The device also has carry propagate and carry generate outputs which may be used for further levels of look-ahead.

The MC2902 is generally used with the bipolar microprocessor unit to provide look-ahead over word lengths of more than four bits. The look-ahead carry generator can be used with binary ALUs in an active LOW or active HIGH input operand mode by reinterpreting the carry functions. The connections to and from the ALU to the look-ahead carry generator are identical in both cases.

The logic equations provided at the outputs are:

$$
\begin{aligned}
C_{n+x} &= G_0 + P_0C_n \\
C_{n+y} &= G_1 + P_1G_0 + P_1P_0C_n \\
C_{n+z} &= G_2 \dot{+} P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_n \\
G &= G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 \\
P &= P_3P_2P_1P_0
\end{aligned}
$$

FEATURES

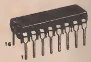
- Provides look-ahead carries across a group of four MC2901 microprocessor ALUs.
- Capability of multi-level look-ahead for high-speed arithmetic operation over large word lengths.
- Typical carry propagation delay of 6.0 ns.

TTL

HIGH-SPEED LOOK-AHEAD CARRY GENERATOR

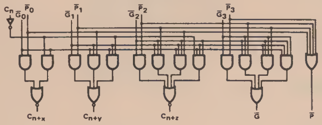
F SUFFIX
CERAMIC PACKAGE
CASE 650

L SUFFIX
CERAMIC PACKAGE
CASE 620

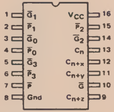


P SUFFIX
PLASTIC PACKAGE
CASE 648

LOGIC DIAGRAM



PIN ASSIGNMENT



ORDERING INFORMATION

Package Type	Temperature Range	Order Number
Molded OIP	0°C to +70°C	MC2902PC
Hermetic OIP	0°C to +70°C	MC2902LC
Hermetic OIP	-55°C to +125°C	MC2902LM
Hermetic Flat Pack	-55°C to +125°C	MC2902FM

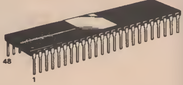
MOTOROLA

MC2903

Product Preview

TTL

FOUR-BIT BIPOLAR MICROPROCESSOR SLICE



FOUR-BIT BIPOLAR MICROPROCESSOR SLICE

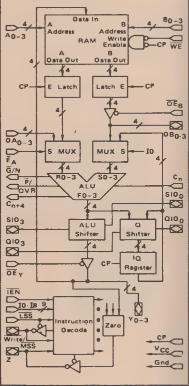
The MC2903 is a four-bit expandable bipolar microprocessor slice. The MC2903 performs all functions performed by the industry standard MC2901A and, in addition, provides a number of significant enhancemants that are especially useful in arithmetic-oriented processors. Infinitely expandable memory and three-port, three-address architecture are provided by the MC2903. In addition to its complete arithmetic and logic instruction set, the MC2903 provides a special set of instructions which facilitate the implementation of multiplication, division, normalization, and other previously time-consuming operations. The MC2903 is supplied in a 48 pin dual in-line package.

- **Built-in Parity Ganaration Circuitry**
  The MC2903 can supply parity across the entire ALU output for use in error detection and CRC code generation.
- **Built-in Sign Extension Circuitry**
  To facilitate operation on different length two's complement numbers, the MC2903 provides the capability to extend the sign at any slice boundary.
- **Expandabla Ragistar Fila**
  Like the MC2901A, the MC2903 contains 16 internal working registers arranged in a two-address architecture. 8ut the MC2903 includes the necessary "hooks" to expand the register file externally to any number of registers.
- **Built-in Multiplication Logic**
  Performing multiplication with the MC2901A requires a few external gates—these gates are contained on-chip in the MC2903. Three special instructions are used for unsigned multiplication, two's complement multiplication, and the last cycle of a two's complement multiplication.
- **Built-in Division Logic**
  The MC2903 contains all logic and interconnects for execution of a non-restoring, multiple-length division with correction of the quotiant.
- **Built-in Normalization Logic**
  The MC2903 can simultaneously shift the Q Register and count in a working register. Thus, the mantissa and exponent of a floating point number can be developed using a single microcycle per shift. Status flags indicate when the operation is complete.

ORDERING INFORMATION

Package Type	Temperature Range	Order Number
Hermetic DIP	0°C to +70°C	MC2903LC
Hermetic DIP	−55°C to +125°C	MC2903LM

BLOCK DIAGRAM



ARCHITECTURE OF THE MC2903

The MC2903 is a high performance, cascadable, 4-bit bipolar microprocessor slice designed for use in CPUs, peripheral controllers, microprogrammable machines, and numerous other applications. The microinstruction flexibility of the MC2903 allows the efficient emulation of almost any digital computing machine. The 9-bit microinstruction selects the ALU sources, function, and destination. The MC2903 is cascadable with full look-ahead or ripple carry, has three-state outputs, and provides various ALU status flag outputs. Advanced low-power Schottky processing is used to fabricate this 48-pin LSI circuit.

All data paths within the device are four bits wide. As shown in the block diagram, the device consists of a 16-word by 4-bit, two-port RAM with latches on both output ports, a high-performance ALU and shifter, a multipurpose Q Register with shifter input, and a 9-bit instruction decoder.

Two-Port RAM

Any two RAM words addressed at the A and B address ports can be read simultaneously at the respective RAM A and B output ports. Identical data appear at the two output ports when the same address is applied to both address ports. The latches at the RAM output ports are transparent when the clock input, CP, is High and they hold the RAM output data when CP is Low. Under control of the $\overline{OE_B}$ three-state output enable, RAM data can be read directly at the MC2903 DB I/O port.

External data at the MC2903 Y I/O port can be written directly into the RAM, or ALU shifter output data can be enabled onto the Y I/O port and entered into the RAM. Data is written into the RAM at the B address when the write enable input, $\overline{WE}$, is Low and the clock input, CP, is Low.

Arithmetic Logic Unit

The MC2903 high performance ALU can perform seven arithmetic and nine logic operations on two 4-bit operands. Multiplexers at the ALU inputs provide the capability to select various pairs of ALU source operands. The E_A input selects either the DA external data input or RAM output port A for use as one ALU operand and the OE_B and I0 inputs select RAM output port 8, DB external data input, or the Q Register content for use as the second ALU operand. Also, during some ALU operations, zeroes are forced at the ALU operand inputs. Thus, the MC2903 ALU can operate on data from two external sources, from an internal and external source, or from two internal sources. Table 1 shows all possible pairs of ALU source operands as a function of the E_A, OE_B, and I0 inputs.

When instruction bits I4, I3, I2, I1, and I0 are Low, the MC2903 executes special functions. Table 4 defines these special functions and the operation which the ALU performs for each. When the MC2903 executes instructions other than the nine special functions, the ALU operation is determined by instruction bits I4, I3, I2, and I1. Table 2 defines the ALU operation as a function of these four instruction bits.

TABLE 1 – ALU OPERAND SOURCES

$\overline{E_A}$	I0	$\overline{OE_B}$	ALU Operand R	ALU Operand B
L	L	L	RAM Output A	RAM Output B
L	L	H	RAM Output A	DB_{0-3}
L	H	X	RAM Output A	Q Register
H	L	L	DA_{0-3}	RAM Output B
H	L	H	DA_{0-3}	DB_{0-3}
H	H	X	DA_{0-3}	O Register

L = Low, H = High, X = don't care.

TABLE 2 – ALU FUNCTIONS

I4	I3	I2	I1	Hex Code	ALU Functions	
L	L	L	L	0	I0 = L	Speciel Functions
					I0 = H	F_i = High
L	L	L	H	1	F = S minus R minus 1 plus C_n	
L	L	H	L	2	F = R minus S minus 1 plus C_n	
L	L	H	H	3	F = R plus S plus C_n	
L	H	L	L	4	F = S plus C_n	
L	H	L	H	5	F = $\overline{S}$ plus C_n	
L	H	H	L	6	F = R plus C_n	
L	H	H	H	7	F = $\overline{R}$ plus C_n	
H	L	L	L	8	F_i = Low	
H	L	L	H	9	$F_i = \overline{R}_i$ AND S_i	
H	L	H	L	A	$F_i = R_i$ EXCLUSIVE NOR S_i	
H	L	H	H	B	$F_i = R_i$ EXCLUSIVE OR S_i	
H	H	L	L	C	$F_i = R_i$ AND S_i	
H	H	L	H	D	$F_i = R_i$ NOR S_i	
H	H	H	L	E	$F_i = R_i$ NAND S_i	
H	H	H	H	F	$F_i = R_i$ OR S_i	

L = Low, H = High, i = 0 to 3.

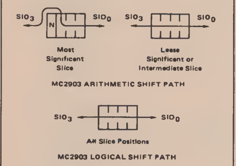
MC2903s may be cascaded in either a ripple carry or lookahead carry fashion. When a number of MC2903s are cascaded, each slice must be programmed to be a most significant slice (MSS), intermediate slice (IS), or least significant slice (LSS) of the array. The carry generate, $\overline{G}$, and carry propagate, $\overline{P}$, signals required for a lookahead carry scheme are generated by the MC2903 and are available as outputs of the least signficant and intermediate slices.

The MC2903 also generates a carry-out signal, C_{n+4}, which is generally available as an output of each slice. Both the carry-in, C_n, and carry-out, C_{n+4}, signals are active High. The ALU generates two other status outputs. These are negative, N, and overflow, OVR. The N output is generally the most significant (sign) bit of the ALU output and can be used to determine positive or negative results. The OVR output indicates that the arithmetic operation being performed exceeds the available two's complement number range. The N and OVR signals are available as outputs of the most significant slice. Thus, the multipurpose $\overline{G}$/N and $\overline{P}$/OVR outputs indicate $\overline{G}$ and $\overline{P}$ at the least significant and intermediate slices, and sign and overflow at the most significant slice. To some extent, the meaning of the C_{n+4}, $\overline{P}$/OVR, and $\overline{G}$/N signals vary with the ALU function being performed. Refer to Table 5 for an exact definition of these four signals as a function of the MC2903 instruction.

ALU Shifter

Under instruction control, the ALU shifter passes the ALU output (F) non-shifted, shifts it up one bit position (2F), or shifts it down one bit position (F/2). Both arithmetic and logical shift operations are possible. An arithmetic shift operation shifts data around the most significant (sign) bit position of the most significant slice, and a logical shift operation shifts data through this bit position (see Figure A). SIO_0 and SIO_3 are bidirectional serial shift inputs/outputs. During a shift-up operation, SIO_0 is generally a serial shift input and SIO_3 a serial shift output. During a shift-down operation, SIO_3 is generally a serial shift input and SIO_0 a serial shift output.

To some extent, the meaning of the SIO_0 and SIO_3 signals is instruction dependent. Refer to Tables 3 and 4 for an exact definiction of these pins.



MC2903 ARITHMETIC SHIFT PATH

MC2903 LOGICAL SHIFT PATH

FIGURE A

The ALU shifter also provides the capability to sign extend at slice boundaries. Under instruction control, the SIO_0 (sign) input can be extended through Y0, Y1, Y2, Y3, and propagated to the SIO_3 output.

A cascadable, 5-bit parity generator/checker is designed into the MC2903 ALU shifter and provides ALU error detection capability. Parity for the F0, F1, F2, F3 ALU outputs and SIO_3 input is generated and, under instruction control, is made available at the SIO_0 output. Refer to the MC2903 applications section for a more detailed description of the MC2903 sign extension and parity generation/checking capability.

The instruction inputs determine the ALU shifter operation. Table 4 defines the special functions and the operation the ALU shifter performs for each. When the MC2903 executes instructions other than the nine special functions, the ALU shifter operation is determined by instruction bits I8, I7, I6, I5. Table 3 defines the ALU shifter operations as a function of these four bits.

Q Register

The Q Register is an auxiliary 4-bit register which is clocked on the Low-to-High transition of the CP input. It is intended primarily for use in multiplication and division operations; however, it can also be used as an accumulator or holding register for some applications. The ALU output, F, can be loaded into the Q Register, and/or the Q Register can be selected as the source for the ALU S operand. The shifter at the input to the Q Register provides the capability to shift the Q Register contents up one bit position (2Q) or down one bit position (Q/2). Only logical shifts are performed. QIO_0 and QIO_3 are bidirectional shift serial inputs/outputs. During a Q Register shift-up operation, QIO_0 is a serial shift input and QIO_3 is a serial shift output. During a shift-down operation, QIO_3 is a serial shift inputs and QIO_0 is a serial shift ouptut.

Double-length arithmetic and logical shifting capability is provided by the MC2903. The double-length shift is performed by connecting QIO_3 of the most significant slice to SIO_0 of the least significant slice, and executing an instruction which shifts both the ALU output and the Q Register.

The Q Register and shifter are controlled by the instruction inputs. Table 4 defines the MC2903 special functions and the operations which the Q Register and shifter perform for each. When the MC2903 executes instructions other than the nine special functions, the Q Register and shifter operation is controlled by instruction bits I8, I7, I6, I5. Table 3 defines the Q Register and shifter operation as a function of these four bits.

TABLE 3 – ALU OESTINATION CONTROL FOR I0 OR I1 QR I2 OR I3 QR I4 = High, $\overline{IEN}$ = Low

					SIQ$_3$		Y3		Y2							
I8	I7	I6	I5	Hex Code	ALU Shifter Function	Most Sig. Slice	Other Slices	Most Sig. Slice	Other Slices	Most Sig. Slice	Other Slices	Y1	Y0	SIO$_0$	$\overline{Write}$	O Reg and Shifter Function	QIO$_3$	QIQ$_0$
L	L	L	L	0	Arith F/2 → Y	Input	Input	F3	SIO$_3$	SIQ$_3$	F3	F2	F1	F0	L	Hold	Hi-Z	Hi-Z
L	L	L	H	1	Log F/2 → Y	Input	Input	SIQ$_3$	SIO$_3$	F3	F3	F2	F1	F0	L	Hold	Hi-Z	Hi-Z
L	L	H	L	2	Arith F/2 → Y	Input	Input	F3	SIO$_3$	SIO$_3$	F3	F2	F1	F0	L	Log O/2 → O	Input	Q0
L	L	H	H	3	Log F/2 → Y	Input	Input	SIO$_3$	SIO$_3$	F3	F3	F2	F1	F0	L	Log Q/2 → O	Input	Q0
L	H	L	L	4	F → Y	Input	Input	F3	F3	F2	F2	F1	F0	Parity	L	Hold	Hi-Z	Hi-Z
L	H	L	H	5	F → Y	Input	Input	F3	F3	F2	F2	F1	F0	Parity	H	Log Q/2 → O	Input	Q0
L	H	H	L	6	F → Y	Input	Input	F3	F3	F2	F2	F1	F0	Parity	H	F → Q	Hi-Z	Hi-Z
L	H	H	H	7	F → Y	Input	Input	F3	F3	F2	F2	F1	F0	Parity	L	F → Q	Hi-Z	Hi-Z
H	L	L	L	8	Arith 2F → Y	F2	F3	F3	F2	F1	F1	F0	SIQ$_0$	Input	L	Hold	Hi-Z	Hi-Z
H	L	L	H	9	Log 2F → Y	F3	F3	F2	F2	F1	F1	F0	SIQ$_0$	Input	L	Hold	Hi-Z	Hi-Z
H	L	H	L	A	Arith 2F → Y	F2	F3	F3	F2	F1	F1	F0	SIO$_0$	Input	L	Log 2O → Q	O3	Input
H	L	H	H	8	Log 2F → Y	F3	F3	F2	F2	F1	F1	F0	SIQ$_0$	Input	L	Log 2Q → O	O3	Input
H	H	L	L	C	F → Y	F3	F3	F3	F3	F2	F2	F1	F0	Hi-Z	H	Hold	Hi-Z	Hi-Z
H	H	L	H	D	F → Y	F3	F3	F3	F3	F2	F2	F1	F0	Hi-Z	H	Log 2O → O	O3	Input
H	H	H	L	E	SIQ$_0$→Y0,Y1,Y2,Y3	SIQ$_0$	SIQ$_0$	SIO$_0$	SIQ$_0$	SIO$_0$	SIO$_0$	SIQ$_0$	SIO$_0$	Input	L	Hold	Hi-Z	Hi-Z
H	H	H	H	F	F → Y	F3	F3	F3	F3	F2	F2	F1	F0	Hi-Z	L	Hold	Hi-Z	Hi-Z

Parity = F3 ⊻ F2 ⊻ F1 ⊻ F0 ⊻ SIQ$_3$

⊻ = Exclusive QR

L = Low H = High Hi-Z = High Impedancē

Output Buffers

The DB and Y ports are bidirectional I/O ports driven by three-stete output buffers with externel output eneble controls. The Y output buffers are enebled when the $\overline{OE_Y}$ input is Low end ere in the high-impedance stete when $\overline{OE_Y}$ is High. Likewise, the DB output buffers are enabled when the $\overline{OE_B}$ input is Low end in the high-impedance stete when $\overline{OE_B}$ is High.

The zero, Z, pin is an open collector input/output that can be wire-ORed between slices. As an output it can be used as e zero detect status flag end generally indicates that the Y0–3 pins are ell Low, whether they ere driven from teh Y output buffers or from en external source connected to the Y0–3 pins. To some extent the meaning of this signal veries with the instruction being performed. Refer to Table 5 for en exact definition of this signal es a function of the MC2903 instruction.

Instruction Decoder

The Instruction Decoder generetes required internal control signals as a function of the nine instruction inputs, I0–8: the Instruction Eneble input, $\overline{IEN}$, the $\overline{LSS}$ input, end the $\overline{Write}/\overline{MSS}$ input/output.

The $\overline{Write}$ output is Low when an instruction which writes date into the RAM is being executed. Refer to Tables 3 and 4 for e definition of the $\overline{Write}$ output es e function of the MC2903 instruction inputs.

When $\overline{IEN}$ is High, the $\overline{Write}$ output is forced High and the Q Register and Sign Compare Flip-Flop contents are preserved. When $\overline{IEN}$ is Low, the $\overline{Write}$ output is enabled and the Q Register end Sign Compare Flip-Flop can be written according to the MC2903 instruction. The Sign Compare Flip-Flop is an on-chip flip-flop which is used during en MC2903 divide operation (see Figure B).

Programming the MC2903 Slice Position

Tying the $\overline{LSS}$ input Low progrems the slice to operete as a least signficant slice (LSS) and enables the $\overline{Write}$ output signal onto the $\overline{Write}/\overline{MSS}$ bidirectional I/O pin. When $\overline{LSS}$ is tied High, tha $\overline{Write}/\overline{MSS}$ pin becomes en input pin; tying the $\overline{Write}/\overline{MSS}$ pin High progrems the slice to operate as en intermediete slice (IS) end tying it Low programs the slice to operete as e most significant slice (MSS).

TABLE 4. SPECIAL FUNCTIONS: I0 = I1 = I2 = I3 = I4 = Low, $\overline{\text{IEN}}$ = Low

					SIO3						
I8 I7 I6 I5	Hex Code	Speciel Function	ALU Function	ALU Shifter Function	Most Sig. Slice	Other Slices	SIO_0	Q Reg end Shifter Function	QIO_3	OIO_0	$\overline{\text{Write}}$
L L L L	0	Unsigned Multiply	F = S + C_n if Z = L F = R + S + C_n if Z = H	Log F/2 → Y (Note 1)	Hi-Z	Input	F0	Log D/2 → O	Input	00	L
L L H L	2	Two's Complement Multiply	F = S + C_n if Z = L F = R + S + C_n if Z = H	Log F/2 → Y (Note 2)	Hi-Z	Input	F0	Log O/2 → Q	Input	Q0	L
L H L L	4	Increment by One or Two	F = S + 1 + C_n	F → Y	Input	Input	Perity	Hold	Hi-Z	Hi-Z	L
L H L H	5	Sign/Magnitude-Two's Complement	F = S + C_n if Z = L F = $\overline{S}$ + C_n if Z = H	F → Y (Note 3)	Input	Input	Perity	Hold	Hi-Z	Hi-Z	L
L H H L	6	Two's Complement Multiply, Last Cycle	F = S + C_n if Z = L F = S − R − 1 + C_n if Z = H	Log F/2 → Y (Note 2)	Hi-Z	Input	F0	Log O/2 → O	Input	00	L
H L L L	8	Single Length Normalize	F = S + C_n	F → Y	F3	F3	Hi-Z	Log 2D → O	03	Input	L
H L H L	A	Double Length Normalize end First Oivide Op	F = S + C_n	Log 2F → Y	R3 ∀ F3	F3	Input	Log 2O → O	03	Input	L
H H L L	C	Two's Complement Oivide	F = S + R + C_n if Z = L F = S − R − 1 + C_n if Z = H	Log 2F → Y	$\overline{\text{R3 ∀ F3}}$	F3	Input	Log 2O → O	03	Input	L
H H H L	E	Two's Complement Divide, Correction end Remainder	F = S + R + C_n if Z = L F = S − R − 1 + C_n if Z = H	F → Y	F3	F3	Hi-Z	Log 2O → O	03	Input	L

NOTES 1. At the most significent slice only, the C_{n+4} signal is internally gated to the Y3 output

2. At the most significent slice only, F3 ∀ DVR is internelly gated to the Y3 output.

3. At the most significent slide only, S3 ∀ F3 is generated et the Y3 output.

4. Op codes 1, 3, 7, 9, 8, O, end F ere reserved for future use

L = Lnw, H = High, X = Don't Care, Hi-Z = High Impedance, ∀ = Exclusive OR, Perity = SIO_3 ∀ F3 ∀ F2 ∀ F1 ∀ F0

MC2903 SPECIAL FUNCTIONS

The MC2903 provides nine special functions which facilitate the implementation of the following operations:

- Single- and double-length normalization
- Two's complement division
- Unsigned and two's complement multiplication
- Conversion between two's complement and sign/magnitude representation
- Incrementation by one or two

Table 4 defines these special functions.

The single-length and double-length normalization functions can be used to adjust a single-precision or double-precision floating point number in order to bring its mantissa within a specified range.

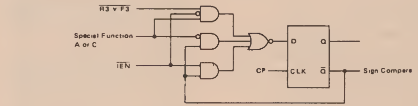
Three special functions which can be used to perform a two's complement, non-restoring divide operation are provided by the MC2903. These functions provide both single- and double-precision divide operations and can be performed in "n" clock cycles, where n is the number of bits in the quotient.

The unsigned multiply special function and the two two's complement multiply special functions can be used to multiply two n-bit, unsigned or two's complement numbers, respectively, in n clock cycles. These functions utilize the conditional add and shift algorithm During the last cycle of the two's complement multiplication, a conditional subtraction, rather. than addition, is performed because the sign bit of the multiplier carries negative weight.

The sign/magnitude-two's complement special function can be used to convert number representation systems. A number expressed in sign/magnitude representation can be converted to the two's complement representation, and vice-versa, in one clock cycle.

The increment-by-one-or-two special function can be used to increment an unsigned or two's complement number by one or two. This is useful in 16-bit word, byte-addressable machines, where the word addresses are multiples of two.

Refer to MC2903 applications section for a more detailed description of these special functions.



The sign compare signal appears at the Z output of the most significant slice during special functions C, D and E, F Refer to Table 5.

FIGURE B – SIGN COMPARE FLIP-FLOP

TABLE 5 – MC2903 STATUS OUTPUTS

Hex I8 I7 I6 I5	Hex I4 I3 I2 I1	I0	Gi (i = 0 to 3)	Pi (i = 0 to 3)	C_{n+4}	$\overline{P}$/OVR		$\overline{G}$/N		Z		
						Most Sig. Slice	Other Slices	Most Sig. Slice	Other Slices	Most Sig. Slice	Intermediate Slice	Least Sig. Slice
X	0	H	0	1	0	0	0	F3	$\overline{G}$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$
X	1	X	$\overline{R}_i \wedge S_i$	$\overline{R}_i \vee S_i$	$G \vee PC_n$	$C_{n+3} \forall C_{n+4}$	$\overline{P}$	F3	$\overline{G}$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$
X	2	X	$R_i \wedge \overline{S}_i$	$R_i \vee \overline{S}_i$	$G \vee PC_n$	$C_{n+3} \forall C_{n+4}$	$\overline{P}$	F3	$\overline{G}$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$
X	3	X	$R_i \wedge S_i$	$R_i \vee S_i$	$G \vee PC_n$	$C_{n+3} \forall C_{n+4}$	$\overline{P}$	F3	$\overline{G}$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$
X	4	X	0	S_i	$G \vee PC_n$	$C_{n+3} \forall C_{n+4}$	$\overline{P}$	F3	$\overline{G}$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$
X	5	X	0	$\overline{S}_i$	$G \vee PC_n$	$C_{n+3} \forall C_{n+4}$	$\overline{P}$	F3	$\overline{G}$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$
X	6	X	0	R_i	$G \vee PC_n$	$C_{n+3} \forall C_{n+4}$	$\overline{P}$	F3	$\overline{G}$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$
X	7	X	0	$\overline{R}_i$	$G \vee PC_n$	$C_{n+3} \forall C_{n+4}$	$\overline{P}$	F3	$\overline{G}$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$
X	8	X	0	1	0	0	0	F3	$\overline{G}$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$
X	9	X	$\overline{R}_i \wedge S_i$	1	0	0	0	F3	$\overline{G}$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$
X	A	X	$R_i \wedge S_i$	$R_i \vee S_i$	0	0	0	F3	$\overline{G}$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$
X	B	X	$\overline{R}_i \wedge S_i$	$\overline{R}_i \vee S_i$	0	0	0	F3	$\overline{G}$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$
X	C	X	$R_i \wedge S_i$	1	0	0	0	F3	$\overline{G}$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$
X	D	X	$\overline{R}_i \wedge \overline{S}_i$	1	0	0	0	F3	$\overline{G}$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$
X	E	X	$R_i \wedge S_i$	1	0	0	0	F3	$\overline{G}$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$
X	F	X	$\overline{R}_i \wedge \overline{S}_i$	1	0	0	0	F3	$\overline{G}$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$
0	0	L	0 if Z = L $R_i \wedge S_i$ if Z = H	S_i if Z = L $R_i \vee S_i$ if Z = H	$G \vee PC_n$	$C_{n+3} \forall C_{n+4}$	$\overline{P}$	F3	$\overline{G}$	Input	Input	Q0
2	0	L	0 if Z = L $R_i \wedge S_i$ if Z = H	S_i if Z = L $R_i \vee S_i$ if Z = H	$G \vee PC_n$	$C_{n+4} \forall C_{n+4}$	$\overline{P}$	F3	$\overline{G}$	Input	Input	Q0
4	0	L	See Note 1	See Note 2	$G \vee PC_n$	$C_{n+3} \forall C_{n+4}$	$\overline{P}$	F3	$\overline{G}$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$	$\overline{Y}0\overline{Y}1\overline{Y}2\overline{Y}3$
5	0	L	0	S_i if Z = L $\overline{S}_i$ if Z = H	$G \vee PC_n$	$C_{n+3} \forall C_{n+4}$		F3 if Z = L F3 ∀ S3 if Z = H	$\overline{G}$	S3	Input	Input
6	0	L	0 if Z = L $\overline{R}_i \wedge S_i$ if Z = H	S_i if Z = L $\overline{R}_i \vee S_i$ if Z = H	$G \vee PC_n$	$C_{n+3} \forall C_{n+4}$	$\overline{P}$	F3	$\overline{G}$	Input	Input	Q0
8	0	L	0	S_i	See Note 3	Q2 ∀ Q1	$\overline{P}$	Q3	$\overline{G}$	$\overline{Q}0\overline{Q}1\overline{Q}2\overline{Q}3$	$\overline{Q}0\overline{Q}1\overline{Q}2\overline{Q}3$	$\overline{Q}0\overline{Q}1\overline{Q}2\overline{Q}3$
A	0	L	0	S_i	See Note 4	F2 ∀ F1	$\overline{P}$	F3	$\overline{G}$	See Note 5	See Note 5	See Note 5
C	0	L	$R_i \wedge S_i$ if Z = L $\overline{R}_i \wedge S_i$ if Z = H	$R_i \vee S_i$ if Z = L $\overline{R}_i \vee S_i$ if Z = H	$G \vee PC_n$	$C_{n+3} \forall C_{n+4}$	$\overline{P}$	F3	$\overline{G}$	Sign Compare FF Output	Input	Input
E	0	L	$R_i \wedge S_i$ if Z = L $\overline{R}_i \wedge S_i$ if Z = H	$R_i \vee S_i$ if Z = L $\overline{R}_i \vee S_i$ if Z = H	$G \vee PC_n$	$C_{n+3} \forall C_{n+4}$	$\overline{P}$	F3	$\overline{G}$	Sign Compare FF Output	Input	Input

NOTES 1 If $\overline{LSS}$ is Low, G0 = S0 and G1,2,3 = 0 If $\overline{LSS}$ is High, G0,1,2,3 = 0

2 If $\overline{LSS}$ is Low, P0 = 1 end P1,2,3 = S1,2,3 If $\overline{LSS}$ is High, $P_i = S_i$

3 At the most significant slice, $C_{n+4} = Q3 \forall Q2$ At other slices, $C_{n+4} = G \vee PC_n$.

4 At the most significant slica, $C_{n+4} = F3 \forall F2$ At other slices, $C_{n+4} = G \vee PC_n$.

5 $Z = \overline{Q}0\overline{Q}1\overline{Q}2\overline{Q}3\overline{F}0\overline{F}1\overline{F}2\overline{F}3$

L = Low = 0. H = High = 1. ∨ = OR. ∧ = AND. ∀ = Exclusive OR

P = P3P2P1P0

G = G3 ∨ G2P3 ∨ G1P2P3 ∨ G0P1P2P3

C_{n+3} = G2 ∨ G1P2 ∨ G0P1P2 ∨ C_n P0P1P2

PIN DEFINITIONS

A0–3 Four RAM address inputs which contain the address of the RAM word appearing at the RAM A output port.

B0–3 Four RAM address inputs which contain the address of the RAM word appearing at the RAM B output port and into which new data is written whan tha $\overline{WE}$ input and the CP input ara Low.

$\overline{WE}$ The RAM write enable input. If $\overline{WE}$ is Low, data at the Y I/O port is written into the RAM when the CP input is Low. When $\overline{WE}$ is High, writing data into the RAM is inhibited.

DA_{0-3} A four-bit external data input which can be selected as one of the MC2903 ALU operand sourcas; DA_0 is the least significant bit.

$\overline{EA}$ A control input which, when High, selects DA_{0-3}, and, when Low, selects RAM output A as the ALU R operand.

DB_{0-3} A four-bit external data input/output. Under control of the $\overline{OE_B}$ input, RAM output port B can be directly read on these lines, or input data on these lines can be selected as tha ALU S operand.

$\overline{OE_B}$ A control input which, when Low, enables RAM output B onto the DB_{0-3} lines and, when High, disables the RAM output B three-state buffers.

C_n The carry-in input to the MC2903 ALU.

I0–8 The nine instruction inputs used to select the MC2903 operation to be performed.

$\overline{IEN}$ The instruction enable input which, when Low, enables the $\overline{Write}$ output and allows the Q Register and the Sign Compare flip-flop to be written. When $\overline{IEN}$ is High, the $\overline{Write}$ output is forced High and the Q Register and Sign Compare flip-flop are in the hold mode.

C_{n+4} This output generally indicates the carry-out of the MC2903 ALU. Refer to Table 5 for an exact definition of this pin.

$\overline{G}$/N A multipurpose pin which indicates the carry generate, $\overline{G}$, function at the least significant and intermediate slices, and generally indicates the sign, N, of the ALU result at the most significant slice. Refer to Table 5 for an exact definition of this pin.

$\overline{P}$/OVR A multipurpose pin which indicatas the carry propagate, $\overline{P}$, function at the least significant and intermediate slices, and indicates the conventional two's complament overflow, OVR, signal at tha most significant slice. Refer to Table 5 for an exact definition of this pin.

Z An open-collector input/output pin which, when High, generally indicatas the Y0–3 outputs are all Low. For soma special functions, Z is used as an input pin. Refer to Table 5 for an exact definition of this pin.

SIO_0, SIO_3 Bidirectional serial shift inputs/outputs for the ALU shifter. During a shift-up operation, SIO_0 is an input and SIO_3 an output. During a shift-down operation, SIO_3 is an input and SIO_0 is an output. Refer to Tablas 3 and 4 for an exact definition of these pins.

QIO_0, QIO_3 Bidirectional serial shift inputs/outputs for the Q shifter which operate like SIO_0 and SIO_3. Refer to Tables 3 and 4 for an exact definition of these pins.

$\overline{LSS}$ An input pin which, when tied Low, programs the chip to act as the least significant slice (LSS) of an MC2903 array and enables the $\overline{Write}$ output onto the $\overline{Write}/\overline{MSS}$ pin. When $\overline{LSS}$ is tied High, the chip is programmad to operate as either an intermediate or most significant slica and the $\overline{Write}$ output buffer is disabled.

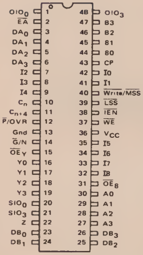
$\overline{Write}/\overline{MSS}$ When $\overline{LSS}$ is tied Low, the $\overline{Write}$ output signal appeacs at this pin; the $\overline{Write}$ signal is Low when an instruction which writes data into the RAM is being executed. When $\overline{LSS}$ is tied High, $\overline{Write}/\overline{MSS}$ is an input pin; tying it High programs the chip to operate as an intermediate slice (IS) and tying it Low programs the chip to operate as the most significant slice (MSS).

Y0–3 Four data inputs/outputs of the MC2903. Under control of the $\overline{OE_Y}$ input, the ALU shifter output data can be enabled onto these lines, or these lines can be used as data inputs when external data is written directly into the RAM.

$\overline{OE_Y}$ A control input which, when Low, enables the ALU shifter output data onto the Y0–3 lines and, when High, disables the Y0–3 three-state output buffers.

CP The clock input to the MC2903. The Q Register and Sign Compare flip-flop are clocked on the Low-to-High transition of the CP signal. When enabled by $\overline{WE}$, data is written in the RAM whan CP is Low.

PIN ASSIGNMENT



MOTOROLA

MC2909
MC2911

TTL
MICROPROGRAM SEQUENCER

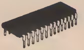
MICROPROGRAM SEQUENCER

The MC2909 is a four-bit-wide address controller intended for sequencing through a series of microinstructions contained in a ROM or PROM. Two MC2909s may be interconnected to generate a twelve-bit address (4K words).

The MC2909 can select an address from any of four sources. They are: 1) a set of external direct inputs (D); 2) external data from the R inputs, stored in an internal register; 3) a four-word-deep push/pop stack; or 4) a program counter register (which usually contains the last address plus one). The push/pop stack includes certain control lines so that it can efficiently execute nested subroutine linkages. Each of the four outputs can be OR'ed with an external input for conditional skip or branch instructions, and a separate line forces the outputs to all zeroes. The outputs are three-state.

The MC2911 is an identical circuit to the MC2909, except the four OR inputs are removed and the D and R inputs are tied together. The MC2911 is in a 20-pin, 0.3" centers package.

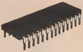
- 4-Bit Slice Cascadable to Any Number of Microwords
- Internal Address Register
- Branch Input for N-Way Branches
- Cascadable 4-Bit Microprogram Counter
- 4 X 4 File with Stack Pointer and Push/Pop Control for Nesting Microsubroutines
- Zero Input for Returning to the Zero Microcode Word
- Individual OR Input for Each Bit for Branching to Higher Microinstructions
- Three-State Outputs
- All Internal Registars Change State on the Low-to-High Transition of the Clock



P SUFFIX
PLASTIC PACKAGE
CASE 710

L SUFFIX
CERAMIC PACKAGE
CASE 732

F SUFFIX
CERAMIC PACKAGE
CASE 737



L SUFFIX
CERAMIC PACKAGE
CASE 733

P SUFFIX
PLASTIC PACKAGE
CASE 73a

MAXIMUM RATINGS (above which tha useful life mey be impaired)	
Storege Tampereture	-65^{o}C to $+150^{o}$C
Temperature (Ambient) Undar Bies	-55^{o}C to $+125^{o}$C
Supply Voltage to Ground Potential	-0.5 V to $+V_{CC}$ max
OC Input Voltege	-0.5 V to $+7.0$ V
OC Output Current, Into Outputs	30 mA
OC Input Current	-30 mA to $+5.0$ mA

ORDERING INFORMATION

Package Type	Temperatura Renge	MC2909 Order Number	MC2911 Order Number
Molded DIP	0^{o}C to $+70^{o}$C	MC2909PC	MC2911PC
Harmetic OIP	0^{o}C.to $+70^{o}$C	MC2909LC	MC2911LC
Hermetic OIP	-55^{o}C to $+125^{o}$C	MC2909LM	MC2911LM
Hermetic Flet Pek	-55^{o}C to $+125^{o}$C	–	MC2911FM

PIN ASSIGNMENTS

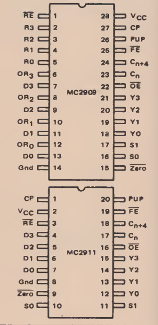
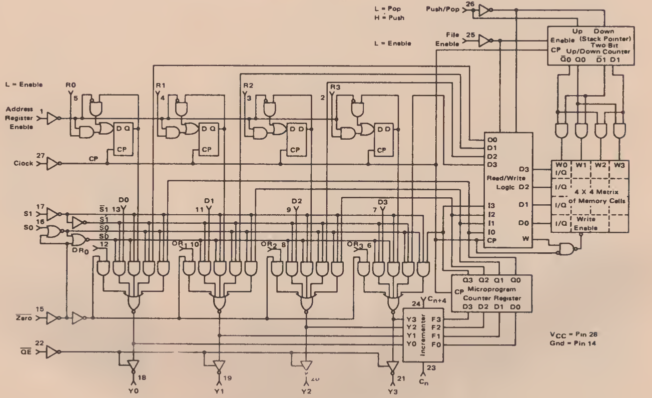
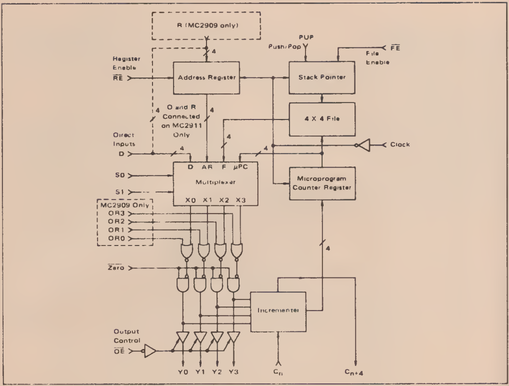


FIGURE 1 – MC2909 MICROPROGRAM SEQUENCER BLOCK DIAGRAM



The MC2911 is an IDENTICAL circuit to the MC2909, except the four DR inputs are removed and the D and R inputs are tied together. The MC2911 is in a 20-pin Dual-in-Line package. See Figure 11.

MICROPROGRAM SEQUENCER



ARCHITECTURE OF THE MC2909/MC2911

The MC2909/MC2911 are bipolar microprogram sequencers intended for use in high-speed microprocessor applications. The device is a cascadable 4-bit slice such that twn devices allow addressing of up to 256-words of microprogram and three devices allow addressing of up to 4K words of microprogram. A detailed logic diagram is shown in Figure 1.

The device contains a four-input multiplexer that is used to select either the address register, direct inputs, microprogram counter, or file as the source of the next microinstruction address. This multiplexer is controlled by the S0 and S1 inputs.

The address register consists of four D type, edge-triggered flip-flops with a common clock enable. When the address register enable is Low, new data is entered into the register on the clock Low-to-High transition. The address register is available at the multiplexer as a source for the next microinstruction address. The direct input is a 4-bit field of inputs to the multiplexer and can be selected as the next microinstruction address On the MC2911, the direct inputs are also used as inputs to the register. This allows an N-way branch where N is any word in the microcode.

The MC2902/MC2911 contains a microprogram counter (μPC) that is composed of a 4-bit incrementer followed by a 4-bit register. The incrementer has carry-in (C_n) and carry-out (C_{n+4}) such that cascading to larger word lengths is straightforward. The μPC can be used in either of two ways. When the least significant carry-in to the incrementer is High, the microprogram register is loaded on the next clock cycle with the current Y output word plus one (Y + 1 → μPC). Thus sequential micro instructions can be executed. If this least significant C_n is Low, the incrementer passes the Y output word unmodified and the microprogram reigster is loaded with the same Y word on the next clock cycle (Y → μPC). Thus, the same microinstruction can be executed any number of times by using the least significant C_n as the control

The last source available at the multiplexer input is the 4 X 4 file (stack). The file is used to provide return address linkage when executing microsubroutines. The file contains a built-in stack pointer (SP) which always points to the last file word written. This allows stack reference operations (looping) to be performed without a push or pop.

The stack pointer operates as an up/down counter with separate push/pop and file enable inputs. When the file enable input is Low and the push/pop input is High, the Push operation is enabled. This causes the stack pointer to increment and the file to be written with the required return linkage—the next microinstruction address following the subroutine jump which initiated the Push.

If the file enable input is Low and the push/pop control is Low, a Pop operation occurs. This implies the usage of the return linkage during this cycle and thus a return from subroutine. The next Low-to-High clock transition causes the stack pointer to decrement. If the file enable is High, no action is taken by the stack pointer regardless of any other input.

The stack pointer linkage is such that any combination of pushes, pops, or stack references can be achieved. One microinstruction subroutine can be performed. Since the stack is four words deep, up to four microsubroutines can be nested.

The $\overline{\text{Zero}}$ input is used to force the four outputs to the binary zero state. When the $\overline{\text{Zero}}$ input is Low, all Y outputs are Low regardless of any other inputs (except $\overline{\text{OE}}$). Each Y output bit also has a separate OR input such that a conditional logic one can be forced at each Y output. This allows jumping to different microinstructions on programmed conditions.

The MC2909/MC2911 feature three-state Y outputs. These can be particularly useful in military designs requiring external Ground Support Equipment (GSE) to provide automatic checkout of the microprocessor. The internal control can be placed in the high-impedance state, and preprogrammed sequences of microinstructions can be executed via external access to the control ROM/PROM.

OPERATION OF THE MC2902/MC2911

Table 4 lists the select codes for the multiplexer. The two bits applied from the microword register (and additional combinational logic for branching) determine which data source contains the address for the next micro-instruction. The contents of the selected source will appear on the Y outputs. Table 4 also shows the truth table for the output control and for the control of the push/pop stack. Table 5 shows in detail the effect of S0, S1, $\overline{\text{FE}}$ and PUP on the MC2909. These four signals define what address appears on the Y outputs and what the state of all the internal registers will be following the clock Low-to-High edge. In this illustration, the microprogram counter is assumed to contain initially some word J, the address register some word K, and the four words in the push/pop stack contain R_a through R_d.

TABLE 4

ADDRESS SELECTION

Octal	S1	S0	Source for Y Outputs	Symbol
0	L	L	Microprogram Counter	μPC
1	L	H	Address Register	AR
2	H	L	Push/Pop Stack	STK0
3	H	H	Direct Inputs	D_i

OUTPUT CONTROL

OR_i	$\overline{\text{Zero}}$	$\overline{\text{OE}}$	Y_i
X	X	H	Z
X	L	L	L
H	H	L	H
L	H	L	Source selected by S0,S1

Z = High Impedence

SYNCHRONOUS STACK CONTROL

$\overline{\text{FE}}$	PUP	Push/Pop Stack Change
H	X	No Change
L	H	Incrementer Stack Pointer, then Push Current PC onto STK0
L	L	Pop Stack (Decrement Stack Pointer)

Table 6 illustrates the execution of a subroutine using the MC2909. The configuration of Figure 3 is assumed. The instruction being executed at any given time is the one contained in the microword register (μWR). The contents of the μWR also controls (indirectly, perhaps) the four signals S0, S1, $\overline{\text{FE}}$, and PUP. The starting address of the subroutine is applied to the D inputs of the MC2909 at the appropriate time.

In the columns on the left is the sequence of microinstructions to be executed. At address J+2, the sequence control portion of the microinstruction contains the command "Jump to the subroutine at A". At the time T2, this instruction is in the μWR and the MC2909 inputs are setup to execute the jump and save the return address. The subroutine address A is applied to the D inputs from the μWR and appears on the Y outputs. The first instruction of the subroutine, I(A), is accessed and is at the inputs of the μWR. On the next clock transition, I(A) is loaded into the μWR for execution, and the return address J+3 is pushed onto the stack. The return instruction is executed at T5. Table 7 is a similar timing chart showing one subroutine linking to a second, the latter consisting of only one microinstruction.

MOTOROLA

MC2910

Product Preview

MICROPROGRAM CONTROLLER

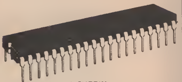
The MC2910 Microprogram Controller is an address sequencer intended for controlling the sequence of execution of microinstructions stored in microprogram memory. Besides the capability of sequential access, it provides conditional branching to any microinstruction within its 4096-microword range. A last-in, first-out stack provides microsubroutine return linkage and looping capability; there are five levels of nesting of microsubroutines. Microinstruction loop count control is provided with a count capacity of 4096.

During each microinstruction, the Microprogram Controller provides a 12-bit address from one of four sources: 1) the microprogram address register, μPC, which usually contains an address one greater than the previous address; 2) an external (direct) input, D; 3) a register/counter, R, retaining data loaded during a previous microinstruction; or, 4) a five-deep last-in, first-out stack, F.

- **Twalva Bits Wida**
  Address up to 4096 words of microcode with one chip. All internal elements are a full 12 bits wide.
- **Intarnal Loop Counter**
  Pre-settable 12-bit down-counter for repeating instructions and counting loop iterations.
- **Four Address Sources**
  Microprogram address may be selected from microprogram counter, branch address bus, five-level push/pop stack, or internal holding register.
- **Sixtaan Powerful Microinstructions**
  Executes 16 sequence control instructions, most of which are conditional on external condition input, state of internal loop counter, or both.
- **Output Enable Controls for Three Branch Address Sources**
  Built-in decoder function to enable external devices onto branch address bus. Eliminates external decoder.
- **All Ragistars Positiva Edge-Triggerad**
  Simplifies timing problems. Eliminates long setup times.
- **Fast Control from Condition Input**
  Delay from condition code input to address output only 27 ns typical.

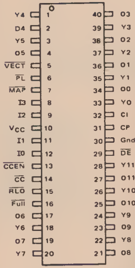
TTL

MICROPROGRAM CONTROLLER



L SUFFIX
CERAMIC PACKAGE
CASE 734

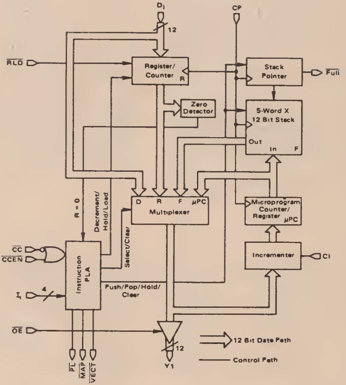
PIN ASSIGNMENT



OROERING INFORMATION

Packege Type	Temperatura Renge	Order Number
Hermetic OIP	0°C to +70°C	MC2910LC
Hermetic DIP	-55°C to +125°C	MC2910LM

FIGURE 1 – BLOCK DIAGRAM



MC2918

QUAD D REGISTER WITH STANDARD AND THREE-STATE OUTPUTS

New Schottky circuits such as the MC2918 register provide the design engineer with additional flexibility in system configuration—especially with regard to bus structure, organization, and speed. The MC2918 is a quadruple D-type register with four standard totem pole outputs and four three-state bus-type outputs. The 16-pin device also features a buffered common clock (CP) and a buffered common output control ($\overline{\text{OE}}$) for the Y outputs. Information meeting the setup and hold requirements on the D inputs is transferred to the Q outputs on the Low-to-High transition of the clock.

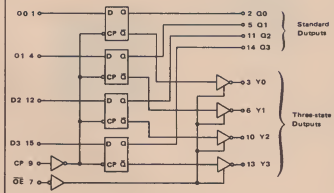
The same data as on the Q outputs is enabled at the three-state Y outputs when the "output control" ($\overline{\text{OE}}$) input is Low. When the $\overline{\text{OE}}$ input is High, the Y outputs are in the high-impedance state.

The MC2918 register can be used in bipolar microprocessor designs as an address register, status register, instruction register or for various data or microword register applications. Because of the unique design of the three-state output, the device features very short propagation delay from the clock to the Q or Y outputs. Thus, system performance and architectural design can be improved by using the MC2918 register. Other applications of MC2918 register can be found in microprogrammed display systems, communication systems and most general or special purpose digital signal processing equipment.

FEATURES

- Advanced Schottky Technology
- Four D-type Flip-flops
- Four Standard Totem Pole Outputs
- Four Three-state Outputs
- 75 MHz Clock Frequency
- 100% Reliability Aśsurance Testing in Compliance With MIL-STD-883

LOGIC DIAGRAM



TTL

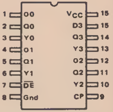
QUAD D REGISTER WITH STANDARD AND THREE-STATE OUTPUTS

F SUFFIX
CERAMIC PACKAGE
CASE 650

L SUFFIX
CERAMIC PACKAGE
CASE 620

P SUFFIX
PLASTIC PACKAGE
CASE 648

PIN ASSIGNMENT



ORDERING INFORMATION

Package Type	Temperature Range	Order Number
Molded DIP	0°C to +70°C	MC2918PC
Hermetic DIP	0°C to +70°C	MC2918LC
Hermetic DIP	−55°C to +125°C	MC2918LM
Hermetic Flat Pack	−55°C to +125°C	MC2918FM

MOTOROLA

MC29100/MC82100
MC29101/MC82101

(DUAL MARKED)

Product Preview

TTL

FIELD PROGRAMMABLE
LOGIC ARRAY

(16 X 8 X 48 FPLA)

FIELD PROGRAMMABLE LOGIC ARRAY

Tha MC29100/MC82100 (three-stata outputs) and the MC29101/MC82101 (opan collector outputs) are bipolar programmable logic errays, containing 48 product tarms (AND tarms), and 8 output functions. Eech output function can be programmed either true active-High (Fp), or true active-Low ($\overset{*}{F}$p). The true stete of the output functions is controlled via an output sum (OR) matrix by a logical combination of 16-input variables, or their complements, up to 48 terms.

Both devices are field-programmable, which means that custom patterns are immediately availabla by following an appropriete fusing procedure.

Tha MC29100 and MC29101 ere fully TTL compatible, and include a chip-eneble clocking input for output de-skewing end inhibit. They feeture either open collector or three-state outputs for eese of expansion of product terms end/or input veriables.

FEATURES

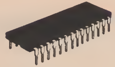
- Fiald Programmable (Ni-Cr Link)
- Input Variables — 16
- Output Functions — 8
- Product Terms — 48
- Address Access Time — 50 ns, Maximum
- Power Dissipation — 600 mW, Typicel
- Input Loading — (−100 μA), Maximum
- Output Option:
  Three-State Outputs —MC29100/MC82100
  Open Collector Outputs — MC29101/MC82101
- Output Disable Function:
  Three-State — Hi-Z
  Opan Collector — Hi
- Ceremic DIP

APPLICATIONS

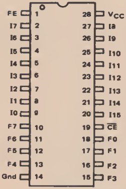
- Large Reed Only Mamory
- Random Logic
- Coda Conversion
- Peripherel Controllers
- Look-Up and Decision Tebles
- Microprogramming
- Address Mapping
- Cherecter Generetors
- Sequential Controllers

This is edvance information end specifications are subject to change without notice.

L SUFFIX
CERAMIC PACKAGE
CASE 733



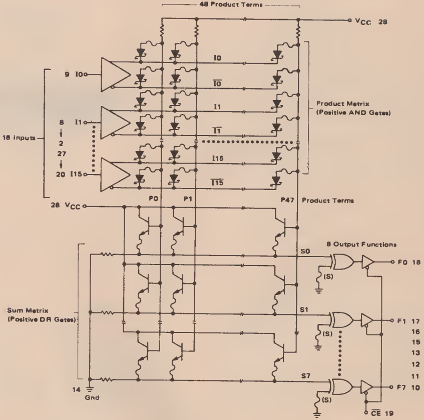
PIN ASSIGNMENT



ORDERING INFORMATION

Temperature Range	Device	Device
0°C to +70°C	MC29100LC MC82100LC	MC29101LC MC82101LC
−55°C to +125°C	MC29100LM MC82100LM	MC29101LM MC82101LM

BLOCK DIAGRAM



TRUTH TABLE

Let $P_n = \Pi_0^{15} (k_m I_m + j_m \overline{I}_m)$; k = 0, 1, X (Don't Care)
n = 0, 1, 2, . . . , 47

where

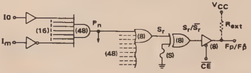
unprogrammed state: $j_m = k_m = 0$

programmed state: $j_m = \overline{k}_m$

$S_r = f(\Sigma_0^{47} P_n)$; $r \equiv p = 0, 1, 2, \ldots, 7$

Mode	P_n	$\overline{CE}$	Fp	$F\bar{p}$	$S_r \overset{?}{=} f(P_n)$
Disabled (MC29101/ MC82101)	X	1	1	1	X
Disabled (MC29100/ MC29100)			Hi-Z	Hi-Z	
Read	1 0	0 0	1 0	0 1	Yes
	X	0	0	1	No

FPLA TYPICAL LOGIC PATH



NOTE: For each of the 8 outputs, either the function Fp (active High) or $F\bar{p}$ (active Low) is available, but not both. The required function polarity is user-programmable via fuse (S).

$P_n = I0\,\overline{I1}\,\overline{I2}\,I3 \ldots \overline{I}_m$

$S_r = P0 + P1 + P2 + \ldots P_n$

$\overline{S}_r = \overline{P}0 \bullet \overline{P}1 \bullet \overline{P}2 \bullet \ldots \overline{P}_n$

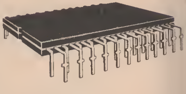
$Fp = (\overline{CE}) + (S_r) = (\overline{CE}) + (P0 + P1 + P2 + \ldots P_n)$ with S = Short

$F\bar{p} = (\overline{CE}) + (\overline{S}_r) = (\overline{CE}) + (\overline{P}0 \bullet \overline{P}1 \bullet \overline{P}2 \bullet \ldots \overline{P}_n)$ with S = Open

MOTOROLA

MC10800

MECL—LSI 4-BIT ALU SLICE



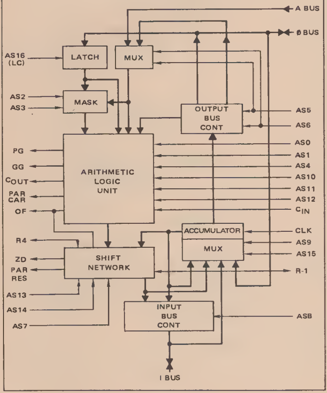
Case 725-01

INTRODUCTION

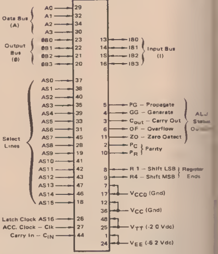
The MC10800 4-Bit ALU Slice is an LSI building block for digital processors. This circuit performs the necessary logic and arithmetic functions required to execute the various machine instructions. Each part is 4 bits wide and is "sliced" parallel to data flow. The MC10800 is fully expandable to larger word lengths by connecting circuits in parallel and features three input/output data ports for maximum system flexibility.

The 4-Bit ALU Slice as shown in the block diagram below contains latch/mask logic, ALU, shift network, accumulator, and bus control logic in a single bipolar circuit. Seventeen select lines are used to control all operations within the part.

4-BIT SLICE BLOCK DIAGRAM



INPUT/OUTPUT DIAGRAM



DS943

TABLE OF CONTENTS

IMPORTANT FEATURES

1. Powerful ALU
   a. Full binary and 8CD arithmetic.
   b. A and Ø input data words treated equal.
   c. All logic functions.
   d. Internal lookahead carry with propagate and generate outputs.
2. Internal shift network.
   a. Left and right logic shift.
   b. Sign protect for arithmetic shift.
3. Versatile bus structure.
4. Master-slave accumulator for temporary storage.
5. Interfaces with MECL 10,000 register file circuits.
6. All necessary status outputs: overflow, zero detect, carry out, and sign bit.
7. Parity outputs for binary operations.
8. Full masking of Ø bus to A bus is provided in the latch/mask network.
9. Each part is 4 bits wide and the circuits can be operated in parallel to form any word size in increments of 4 bits.

M10800 LSI FAMILY DEVICES:

P/N	Description
MC10800	4-Bit ALU Slice
MC10801	Microprogram Control Function
MC10802	Timing Function
MC10803	Memory Interface Function

COMPATIBLE MOTOROLA MECL MEMORIES:

MCM10143	8 x 2 Multiport Register File
MCM10144	256 x 1 RAM
MCM10145	16 x 4 RAM
MCM10146	1024 x 1 RAM
MCM10147	128 x 1 RAM
MCM10149	256 x 4 PROM

COMPATIBLE LOGIC:

ECL 10,000: 100 Circuits, Industry-wide

ABSOLUTE MAXIMUM RATINGS (see Note 1)

RATING	SYMBOL	VALUE	UNIT
Supply Voltage (V_{cc} = 0)	V_{EE} V_{TT}	−8 to 0 −4 to 0	Vdc Vdc
Input Voltage Std (V_{cc} = 0) Bus	V_{in} V_{in}	0 to V_{EE} Note 2	Vdc Vdc
Output Source Cont Current Surge	I_o I_o	< 50 <100	mAdc mAdc
Storage Temp.	$T_{stg.}$	−55 to +150	°C
Junction Temp.	T_j	165	°C

NOTE: 1. Permanent device damage may occur if absolute maximum ratings are exceeded. Functional operation should be restricted to RECOMMENDED OPERATING CONDITIONS. Exposure to higher than recommended voltages for extended periods of time could affect device realiability.

NOTE: 2. Input voltage limit is V_{CC} to −2 Volts when the bus is used as an input and the output drivers are disabled.

SYSTEM OVERVIEW

Certain basic functional building blocks, as shown in Figure 1, are characteristic of high performance processors. These building blocks can be resolved into LSI circuits which, by proper use of control memory programming and circuit function select lines, will fit a wide range of system requirements. The Motorola M10800 family of LSI processor circuits is designed to provide these functional blocks and not limit the final system to any given system size or architecture.

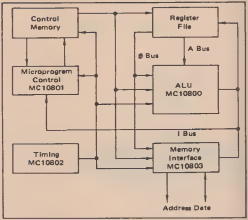
The ALU block in Figure 1 is filled by the MC10800 4-Bit ALU Slice. This circuit combines the mask logic, ALU, shift network, and accumulator to give a very powerful function set. In addition, the data routing paths and data I/O ports allow numerous options when configuring a system. When designing with the MC10800 function set it is possible to accomplish in a single pass what would require mulitple passes with other ALU circuits.

In the M10800 Family, the register file has been made a separate block from the ALU, because modern systems use a wide range of register file sizes — varying between 4 and 256 working registers. .With high speed MECL RAM and register file circuits available, the designer is permitted to specify the optimum register file size and configuration for his particular system. Storage registers are available in the MC10801 Microprogram Control Function and MC10803 Memory Interface Function for special purpose functions such as instruction register, status register, program counter, index register, and stack pointer.

Virtually all modern computers use a microprogrammed instruction set. Microprogramming permits emulating machines, updating systems by increasing capability, modifying systems to meet specific customer requirements, and designing to take advantage of existing software. The MC10801 Microprogram Control Function contains the logic needed to address and sequence through microprogram storage. Each MC10801 is 4 bits wide and can be operated in parallel for larger control memory address words. The necessary storage, logic, and I/O is provided to generate next control memory address and handle status, branching, and interrupt functions.

One of the penalties normally paid to gain the advantages of microprogramming is system speed. Each processor instruction requires several microprogram steps. The Motorola M10800 LSI family makes use of MECL 10,000 circuit technology and interfacing to attain fast microprogram cycle times. In addition, other features of the family (such as a powerful ALU function set in the MC10800 and independent memory addressing in the MC10803), minimize the number of microprogram steps per system instruction. With the M10800 bipolar LSI family, it is possible to build fast microprogrammed systems which outperform dedicated hardwired systems using a slower technology.

FIGURE 1 – MICROPROGRAMMED PROCESSOR



The Control Memory block in Figure 1 is a separate section of the system, best selected by the designer. Any microprogram storage included on the LSI circuits results in design constraints. Microprogram storage can vary up to several thousand words, depending on system complexity, and is best built with individual MECL PROMs such as the MCM10149, or MECL RAMs such as the MCM10144 or MCM10146.

Any processor system must have access to external information from such sources as main memory, peripherals, and bulk storage. In the M10800 Family this chore is handled by the M10803 Memory Interface Function. This circuit is 4 bits wide and contains the necessary memory data and address storage. In addition, there are registers and an ALU for performing the various modes of memory addressing.

The MC10802 Timing Function ties the other function blocks together. This part provides the various clock phases as needed and makes it easy to interface to a manual test or control panel. As with other parts in the M10800 Family, the MC10802 is fully programmable for maximum system flexibility.

The Motorola M10800 circuits interface directly with all parts in the MECL 10,000 family. This provides a source for high speed ECL memories and interface circuits for MOS memories. It allows special hardware functions to be constructed for maximum system performance. MECL 10,000 MSI circuits can be used to multiplex status inputs for branch conditions, format priority interrupts, and build high speed array multipliers.

Versatility is a main point of the M10800 Family. The block diagram in Figure 1 is intended to illustrate the purpose of the various LSI functions and not restrict the designer to any particular system configuration or application.

PIN ASSIGNMENTS

Pin Designation	Pin Number	Description
A0	29	Data Bus A — LSB Input
A1	32	Data Bus A — NLSB Input
A2	34	Data Bus A — NMSB Input
A3	30	Data Bus A — MSB Input
ØB0	23	Output Bus — LSB I/O
ØB1	22	Output Bus — NLSB I/O
ØB2	21	Output Bus — NMSB I/O
ØB3	20	Output Bus — MSB I/O
IB0	13	Input Bus — LSB I/O
IB1	14	Input Bus — NLSB I/O
IB2	15	Input Bus — NMSB I/O
IB3	16	Input Bus — MSB I/O
AS0	37	Y Input Mux — Select Input
AS1	3B	Y Input Mux — Select Input
AS4	39	Increment/Decrement by 2 — Select Input
AS2	40	X Input Mux — Select Input
AS3	35	X Input Mux — Select Input
AS5	33	Output Bus Control & A Input Mux — Select Input
AS6	31	Output Bus Control & A Input Mux — Select Input
AS10	41	Add/Subtract — Select Input
AS11	42	Binary/BCD — Select Input
AS12	43	Arithmetic/Logic Mode — Select Input
C_{in}	44	Carry Input
C_{out}	3	Carry Output
PG	5	Group Propagate Output
GG	4	Group Generate Output
OF	6	Overflow Output
PC	2	Parity of Carries Output
PR	10	Parity of Result Output
ZD	11	Zero Detect
AS7	45	Shift Network — Source Select Input
AS13	47	Shift Network — Functidn Select Input
AS14	46	Shift Network — Function Select Input
R4	9	Shift Network — MSB I/O
R-1	8	Shift Network — LSB I/O
AS9	19	Accumulator Mux & Input Bus Control — Select Input
AS15	1B	Accumulator Mux & Input Bus Control — Select Input
ASB	28	Input Bus Driver — Enable Input
CLK	27	Accumulator — Clock Input
AS16 (LC)	26	Output Bus Latch — Clock Input
V_{EE}	1	−5.2 Volt Supply
V_{EE}	24	−5.2 Volt Supply
V_{TT}	25	−2.0 Volt Supply
V_{TT}	4B	−2.0 Volt Supply
V_{CC}	12	Ground
V_{CC}	36	Ground
V_{CCO}	7	Ground
V_{CCO}	17	Ground

ARCHITECTURAL DESCRIPTION

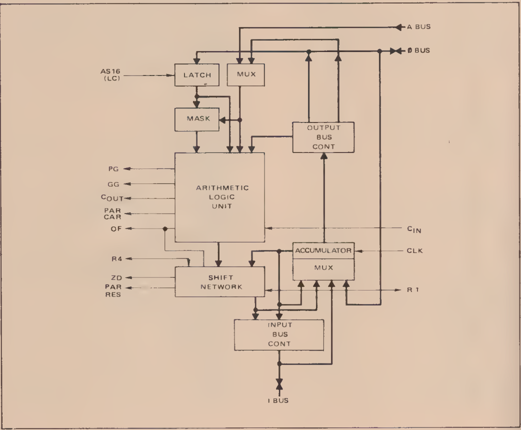
Data enters and exits the MC10800 4-Bit ALU Slice through the A bus, Output (ø bus), and Input (I bus) as shown in Figure 2. The ø bus and I bus are bidirectional, while the A bus is an input port only. These ports are each 4 bits wide requiring that MC10800s be operated in parallel for larger word lengths, single bit data paths C_{in}, C_{out}, R-1, and R4 are used to interconnect parallel MC10800s. The circuit contains two storage elements; the ø bus latch controlled by the latch clock (LC) and the accumulator controlled by the master clock (CLK). The remaining terminals in Figure 2 are status outputs and are used for second level look-ahead carry or for generation of processor condition codes. The individual blocks and I/O terminals in Figure 2 are described below.

Latch/Mask Network

The latch/mask network controls data to one input port of the Arithmetic Logic Unit. The holding latch is positioned to provide temporary storage for data entering through the Output Bus port. The latch clock input (LC) controls the latch operation. When not latched, data ripples through the latch and need not be clocked.

For microprocessor and microcontroller applications, it is desirable to be able to mask data entering a machine. The latch/mask network incorporates this feature. By using the mask select lines, it is possible to mask data on the Output Bus with the A Bus using the logic AND and logic OR functions.

FIGURE 2 – 4-BIT ALU SLICE – MC10800
DATA, STATUS AND CLOCK



Arithmetic Logic Unit

The Arithmetic Logic Unit (ALU) combined with the latch/mask network has the capability of perfoming logic operations, binary arithmetic, and BCD arithmetic on combinations of one, two, or three variables. These variables are the A bus, output bus latch, and accumulator. Variables are treated equally in both binary and BCD formats (A bus minus ø bus and ø bus minus A bus). BCD arithmetic operations are incorporated internally within the ALU and the BCD functions do not compromise system speed.

The ALU incorporates a 9's complement circuit to generate the necessary BCD complement function. The 9's complementer is used with BCD subtract and 9's complement instructions and the circuit is automatically enabled when these functions are selected.

The ALU section of the 4-Bit Slice provides the logic for overflow and carry out. Overflow provides the two's complement overflow of binary addition and subtraction. Overflow is also generated in the shift network and is the exclusive OR of the MSB and NMSB during a shift left operation. Carry out functions for both binary and BCD operations. If second level look-ahead carry is not used, the carry out of one 4-Bit Slice is connected to carry in of the following slice circuit for ripple carry.

The ALU generates the group propagate, group generate, and parity of carry outputs. Group propagate and generate are used for external look-ahead carry between 4-Bit Slice circuits. The propagate and generate outputs operate with both BCD and binary functions. Parity of carries is used for arithmetic error checking and is generated by the exclusive-ORing of C_{in}, carry from the LSB, carry from the NLSB, and carry from the NMSB.

Shift Network

The shift network following the ALU performs the data shift operations within the 4-Bit Slice. Select lines to the shift network control shift left, logic shift right, arithmetic shift right, and ripple through.

The arithmetic shift right provides sign protection for arithmetic shift operations. Only MSB is affected during an arithmetic shift right (towards the LSB), the most significant bit is repeated. R-1 and R4 input/outputs are brought out and are used for shift expansion when interconnecting 4-Bits Slice circuits. The zero detect is derived from the shift network outputs and detects the binary or BCD all zero state. Parity of results is also generated in the shift network. This output, used for parity checking, is generated by exclusive-ORing the shift outputs.

Input Bus Control

The input bus control manipulates the source of data to the Input Bus port. The input bus can receive data from either the shift network or the accumulator. In addition, this control circuit can inhibit data from being routed to the input bus. This allows the input bus to enter data into the accumulator or to be used for other system functions not related to the 4-Bit Slice.

Accumulator/Multiplexer

The master-slave accumulator provides for high speed iterative computer operations. These can include repeated add with accumulated sum, multiply, divide, and multiple shift operations. A multiplexer circuit feeds the accumulator from one of three possible sources as controlled by select lines. These sources are the results of the shift network, the input bus, or the output bus. A fourth condition inhibits the accumulator clock and stored data is retained. Data is entered on the rising (V_{OL} to V_{OH}) clock edge.

Output Bus Control

The output bus control section distributes the outputs of the accumulator to various points in the 4-Bit slice. Select lines route the accumulator to either the A input multiplexer or to the output bus. In addition, the accumulator can be routed to the ALU for mask and compare type operation. A fourth state of the output bus control inhibits the accumulator from going to any of the three above destinations.

A Input

The A input consists of four pins, A0, A1, A2, and A3 which serve as input data paths to the arithmetic logic unit. These inputs are designed to operate in a negative logic data format with a MECL V_{OL} being a logic 1. Because of the BCD functions, the 4-Bit Slice does not directly accept both positive and negative logic formats. The inputs are designated with A0 as the least significant of the 4 bits in the circuit and A3 as the most significant bit.

Output Bus

The output bus consits of four terminals, øB0 through øB3, which function as both data inputs and data outputs. As with the A input, the output bus pins are in negative logic and øB0 is the least significant bit within the part. The output bus when used as an input is routed to the holding latch, and accumulator multiplexer. As an output port, these terminals are used to connect data in the accumulator to the output bus as shown in Figure 2.

Input Bus

The input bus consists of four terminals, IB0 through IB3, which function as both data inputs and data outputs. As with the A buffer and the output bus, the input bus pins are in negative logic and IB0 is the least significant bit within the part. The input bus when used as an input is routed to the accumulator. As an output port, these terminals are used to connect data from either the accumulator or shift network results to the input bus as shown in Figure 2.

Carry In

Carry in, C_{in}, is used to interconnect 4-Bit Slice circuits in a system. For ripple carry, carry in is connected to carry out of the preceding 4-Bit Slice. When look-ahead carry is incorporated, the carry in is connected to the look-ahead carry logic.

Carry in is only used for arithmetic operations and has no effect on any logic operation. The carry in functions for both binary and BCD arithmetic operations. Carry in operates in a negative logic mode with V_{OL} being a logic 1.

Carry Out

Carry out, C_{out}, signals that the calculated value within the ALU has exceeded the maximum capacity of the four ALU output lines. Any binary total over count 15 (1111) or BCD total over count 9 (1001) results in a carry out. When ripple carry is used, carry out is connected to carry in of the following 4-Bit Slice.

Shift Interconnects R-1 and R4

R-1 and R4 are provided to interconnect 4-Bit Slice circutis for shift operations. R-1 and R4 function as both inputs and outputs depending on the shift direction. For a shift left (toward the MSB) R-1 is an input for the R0 bit and R4 is an output for the MSB. For a logic shift right R-1 is an output for the LSB and R4 is an input to R3. MSB is also connected to R4 during a no shift operation and during an arithmetic shift right. This allows R4 to be used as a status output for sign detection. When not used as outputs, the internal drivers for R-1 and R4 are held at a negative logic 1 so the shift interconnects can function as inputs using the MECL emitter dot. See Table 1.

TABLE 1

SHIFT OPERATION	I/O FUNCTION	
	R-1	R4
Shift Left	Shift Input	Shift Output
No Shift	Not Used	MSB Output
Logic Shift Right	Shift Output	Shift Input
Arithmetic Shift Right	Shift Output	MSB Output

Group Propagate and Group Generate

The group propagate, PG, and group generate, GG, outputs are used in conjunction with external look-ahead carry logic for faster system operation. Using this technique, the carry in signals to the 4-Bit Slice circuits are generated faster than with ripple carry. The propagate output goes to the logic 1 when the maximum number value occurs on the ALU outputs. This is count 15 (1111) for binary functions and count 9 (1001) for BCD functions. For binary functions, generate occurs with any value of 16 (10000) or larger and for BCD functions any number value of 10 (10000) or larger.

Group propagate and group generate outputs are used only for arithmetic operations in a system to allow faster generation of carry in signals. They serve no function for ALU logic operations.

Overflow OF

Overflow is used only with two's complement arithmetic and shows that the maximum system word or byte value has been exceeded. In a system, only the overflow output from the 4-Bit Slice operating on the most significant bits of the data word is used.

In addition to overflow caused by an ALU operation, it is possible to have overflow as a result of a shift left (toward the MSB) in the shift network. This happens when the sign bit is changed as a result of the shift left operation.

Normally the overflow of the ALU and shift network are ORed together so that either causes an overflow condition. The exception to this occurs when the accumulator is routed to the shift network inputs. At this time, the ALU overflow is inhibited from the OF output. Overflow is not used with BCD arithmetic.

Zero Detect ZD

Zero detect signals the all zero condition (0000) at the output of the shift network. Zero detect functions for logic operations, binary arithmetic, and BCD arithmetic operations within the ALU. By having the zero detect at the output of the shift network, it is possible to detect zero status after a shift has been performed. Zero detect is defined by the following equation:

$$ZD = \overline{R0}\cdot\overline{R1}\cdot\overline{R2}\cdot\overline{R3}$$

where $\overline{R0}$ through $\overline{R3}$ are the internal outputs from the shift network.

Parity Outputs PAR CAR and PAR RES

Parity bits are used to detect system errors in data handling. With a single parity bit, it is possible to detect a single bit error or any combination of an odd number of bit errors.

For parity checking binary arithmetic operations, two parity points are generated in the MECL 4-Bit Slice. These are parity of carries (PAR CAR) and parity of results (PAR RES). Parity of carries is the parity of the individual bit carries internal to the slice.

$$PAR\ CAR = C\ IN \oplus C0 \oplus C1 \oplus C2$$

Parity of results is the parity of the individual result bits at the output of the shift network.

$$PAR\ RES = R0 \oplus R1 \oplus R2 \oplus R3$$

Accumulator Clock CLK

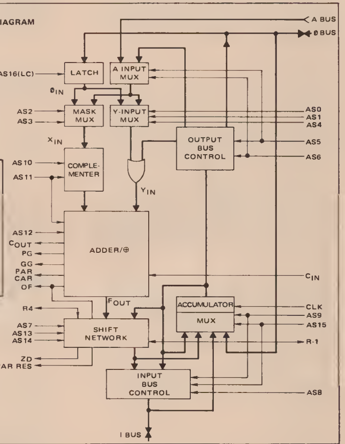
The accumulator is constructed of master-slave flip flops and must be clocked to change stored data. As is characteristic of MECL flip flops, the accumulator is clocked on the positive going (V_{OL} to V_{OH}) clock edge. At that time, data on the accumulator inputs is transferred to the accumulator outputs.

Signals on the accumulator inputs can change at any time with the clock input at either logic state and not change accumulator outputs. The only restriction on changing accumulator inputs is during the set up and hold time near the positive going clock edge.

Latch Clock AS16

Latch clock AS16 controls the storage of data in the holding latch on the output bus. When the latch clock is at V_{IH} data ripples through the latch, interconnecting the output bus with the ALU inputs. When AS16 is at V_{OL} data is stored in the latch and latch outputs are not affected by any changes in information on the output bus.

FIGURE 3 – FUNCTIONAL BLOCK DIAGRAM



FUNCTIONAL DESCRIPTION

Seventeen select lines AS0 through AS16 control the flow of data within the MC10800 4-Bit Slice and determine the arithmetic, logic, and shift operations performed on the data. The following information describes the operation of these select lines, then shows how these select lines combine to perform the various MC10800 functions. *The truth tables are expressed in negative logic with V_{OL} being a logic 1 and V_{OH} a logic 0.* Figure 3 applies.

SELECT LINE OPERATION

Y Input Mux Select Inputs AS0 and AS1

AS0 and AS1 control the source of data to the Y-Input mux of the ALU. These select lines allow selection of either the A Input Mux, the output bus, all logic 0 or all logic 1 bits. Table 2 illustrates the operation of these two select lines.

The ALU Y input is also the port for entering accumulator data into the ALU. This is accomplished by setting AS0 and AS1 to a logic 0 and enabling the accumulator with AS5 and AS6 as described in Table 4 and Table 5.

TABLE 2

AS0	AS1	ALU Y INPUT
0	0	LOGIC 0
0	1	OUTPUT BUS LATCH
1	0	A INPUT MUX
1	1	LOGIC 1

Increment/Decrement by 2 Select Input AS4

Select line AS4 is used to give the MECL 4-Bit Slice an increment or decrement by 2 function capability. When this input is held at a logic 1 (V_{OL}) it has no effect on the circuit. When at a logic 0, it is used with AS0 and AS1 to force a code 0010 (plus 2) or 1110 (minus 2) on the Y input of the ALU. In a system, this input would normally be used only with the 4-Bit Slice operating on the least significant bits in a word or byte. However, other slice locations could also be used to add such constants as 2, 32, 34, 512, 544, 546, etc. AS4 operation with AS0 and AS1 is shown in Table 3.

Output Bus Control and A Input Mux Select Inputs AS5 & AS6

Select lines AS5 and AS6 control the destination of the accumulator output. The accumulator can be routed to five locations in the MECL 4-Bit Slice. Three locations, the output bus, the ALU A input, the ALU ACC input are controlled by AS5 and AS6. A fourth state of AS5 and AS6 inhibits the accumulator from any of these three destination. Other destinations for the accumulator are the input bus as described in the section on AS9 and AS15 and to the shift network inputs as controlled by AS7. (Table 9 and Table 11).

When drivers to the output bus are not enabled by AS5 and AS6, they are forced to a logic 1 (V_{OL}) so this bus can take advantage of the MECL emitter dotting. The accumulator input to the ALU requires a logic 0 when not used as a data input.

AS5$\cdot\overline{AS6}$ enables the accumulator on the ø Bus. The MC10800 can simultaneously output the accumulator contents onto the system ø Bus and input the accumulator contents to the ø Bus port of the ALU. Only external data on the ø Bus is ANDed to the accumulator contents when AS5$\cdot\overline{AS6}$ is selected.

The MC10B00 A bus terminals are input only and AS5 and AS6 select either the A bus inputs or the accumulator to the ALU A input.

Logic State $\overline{AS5}\cdot$AS6 operates in conjunction with AS0 and AS1 to enter data into the ALU Y input. The output of the Y-input mux is logically ORed to the accumulator ALU input. Table 5 illustrates the operation of these select lines.

X Input Mux Select Inputs AS2 and AS3

Select lines AS2 and AS3 control the data path to the other ALU input (X input). These lines can select either the A bus or the output bus. In addition, AS2 and AS3 provide masking capability within the 4-Bit Slice. These select lines control the logic functions — (A bus OR ø bus) and (A bus AND ø bus). This allows any bit or bits to be masked to either a logic 1 or 0 with masking information on either the A bus or ø bus terminals. The advantage of doing masking prior to the ALU is that it allows single pass mask and compare within the 4-Bit Slice. AS2 and AS3 operation is shown in Table 6.

TABLE 3

AS4	ASO	AS1	ALU Y INPUT
1	TABLE 2		OETERMINEO BY ASO, AS1
0	0	0	PLUS 2 (0010)
0	1	1	MINUS 2 (1110)

The combinations AS0$\cdot\overline{AS1}\cdot\overline{AS4}$ and $\overline{AS0}\cdot$AS1$\cdot\overline{AS4}$ are not normally used. AS0$\cdot\overline{AS1}\cdot\overline{AS4}$ results in Y0 = logic 0, Y1 = logic 1, Y2 = A2, end Y3 = A3. $\overline{AS0}\cdot$AS1$\cdot\overline{AS4}$ results in Y0 = logic 0, Y1 = logic 1, Y2 = øB2, and Y3 = øB3.

TABLE 4

AS5	AS6	ø BUS	A IN MUX	ALU
0	0	ø BUS	A BUS	0
0	1	ø BUS	A BUS	ACC
1	0	ACC·ø BUS	A BUS	0
1	1	ø BUS	ACC	0

TABLE 5

$\overline{AS5}\cdot$AS6	ASO	AS1	ALU Y INPUT
0	TABLE 2		Determined by ASO, AS1
1	0	0	ACCUMULATOR
1	0	1	ACC OR ø BUS
1	1	0	ACC OR A MUX
1	1	1	LOGIC 1

TABLE 6

AS2	AS3	ALU X INPUT
0	0	A MUX ANO ø BUS
0	1	ø BUS
1	0	A MUX
1	1	A MUX OR ø BUS

Add/Subtract and Binary/BCD Select Inputs AS10 and AS11

Select line AS10 and add/subtract control enables the complementer. During the add mode the X_{in} data is passed directly through, and during the subtract mode the data is complemented. The complement function is also modified by AS11. The 9's complement is generated for BCD subtract, and data is inverted (1's complement) for binary subtract.

If the ALU is in the logic (Exclusive-OR) mode the complementer is used to selectively invert the X_{in} data. AS11 should be set to the binary mode, and AS10 is used to control inversion of the data.

TABLE 7

AS10	AS11	FUNCTION
0	0	SUBTRACT BCD (9's COMPLEMENT)
0	1	SUBTRACT BINARY (INVERT)
1	0	ADD BCD
1	1	ADD BINARY

Arithmetic/Logic Mode Select Input AS12

AS12 is the mode control for the 4-Bit adder. This input determines if the function performed in the ALU is an arithmetic or logic operation. The logic mode disables the carry between bits and the function performed is the Exclusive-OR of the two inputs to the adder.

TABLE 8

AS12	MODE
0	LOGIC (Exclusive-OR)
1	ARITHMETIC

Shift Network Source Select Input AS7

AS7 controls the information source to the shift network. The MECL 4-Bit Slice is designed to allow shifting data from the accumulator or from the ALU. The accumulator shift operation is useful in multiply and divide add/shift routines. AS7 follows the truth table shown in Table 9.

TABLE 9

AS7	SHIFT NETWORK SOURCE
0	ACCUMULATOR
1	ALU

Shift Network Function Select Inputs AS13 and AS14

AS13 and AS14 control the operation of the shift network following the ALU in the 4-Bit Slice. The four possible operations are: no shift (straight through), shift left one bit, logic shift right one bit, and arithmetic shift right one bit. The truth table for AS13 and AS14 is shown in Table 10.

TABLE 10

AS13	AS14	SHIFT OPERATION
0	0	SHIFT LEFT
1	0	NO SHIFT
0	1	LOGIC SHIFT RIGHT
1	1	ARITHMETIC SHIFT RIGHT

Shift left shifts each bit at the shift network inputs (F inputs) one bit left (toward the MSB). This operation provides the function for both arithmetic and logic shift left. Logic shift right shifts each bit at the F inputs one bit right (toward the LSB). This shift mode is used in all 4-Bit Slice circuits for logic shift right and all except the slice circuit handling the most significant bit of a word or byte for an arithmetic shift right. During an arithmetic right shift, it is necessary to have sign protection for a number expressed in 2's complement or 1's complement notation. This is accomplished by repeating the most significant bit during and arithmetic shift right.

R-1 is an input for shift left and an output for both logic and arithmetic shift right. This pin is not used for no shift. R4 is an input for logic shift right and an output for all other AS13 and AS14 operations. This feature allows R4 to function as a sign bit status output on the MC10800 having the sign bit as the MSB within the part.

Accumulator Mux & Input Bus Control Select Inputs AS9 & AS15

Select lines AS9 and AS15 perform two functions in the MECL 4-Bit Slice. One is to control the source of data to the accumulator, the other to control the source of data to the input bus drivers. The accumulator can store data from three independent points in the 4-Bit Slice. These are the input bus, the output bus, and the shift network outputs. A fourth condition on AS9 and AS15 feeds the accumulator back on itself so the accumulator clock is effectively disabled. This permits storage of data in the accumulator with a continuous system clock entering the slice circuit. The clock disable state of AS9 and AS15 is designed so that only the clock can load information into the accumulator and the accumulator status cannot be altered by the select lines alone.

AS9 and AS15 route either the accumulator or the shift network outputs to the input bus drivers. When the results of the shift network are gated to the accumulator, the accumulator is the source of data to the input bus drivers. For all other combinations AS9 and AS15, the shift network outputs are gated to the bus drivers. The accumulator clock can be disabled when reading the accumulator to the shift network and using the shift network as a feedback path. Table 11 illustrates the operation of AS9 and AS15.

TABLE 11

AS9	AS15	INPUT TO ACCUMULATOR	INPUT BUS SOURCE
0	0	SHIFT RESULTS	ACCUMULATOR
0	1	OUTPUT BUS	SHIFT RESULTS
1	0	INPUT BUS	SHIFT RESULTS
1	1	ACCUMULATOR	SHIFT RESULTS

Input Bus Driver Enable Input AS8

AS8 inhibits and enables the input bus driver circuits. When this select line is at a logic 1 (V_{OL}) the input bus drivers are enabled and data from either the shift network or the accumulator is routed to the input bus. A logic 0 on AS8 disables the input bus drivers so the input hus port can be used to input data or so the input bus can route data independent of the 4-Bit Slice in a system. When disabled the input bus drivers assume a logic 1 state (V_{OL}). Forcing the outputs low permits the use of MECL emitter dotting on the system input bus. The truth table for AS8 is shown in Table 12.

TABLE 12

AS8	INPUT BUS
0	DISABLE OUTPUTS
1	ENABLE OUTPUTS

Accumulator Clock Input CLK and Output Bus Latch Clock Input — AS16

Data is entered into the accumulator on the rising edge of the clock signal. The data source is selected by AS9 and AS15. Latch clock AS16 controls the storage of data in the holding latch on the output bus. When AS16 is a logic 0 (V_{IH}) data ripples through the latch. When AS16 is a logic 1 data is stored in the latch and the latch outputs are not affected by information changes on the output bus. Table 13 is the truth table for AS16.

TABLE 13

AS16	LATCH OPERATION
0	ENABLED
1	LATCHED

ALU LOGIC OPERATION FUNCTION SET

The output bus latch, the A input multiplaxar, and tha accumulator ere sources of data to tha ALU. Following the verious truth tebles of the given select lines e full set of logic operation can ba performed in tha ALU.

The equivelent block diagram of tha ALU for logic operations is shown in Figure 4. Tha addar is set to the logic mode (AS12 = 0), tharafora, F_{OUT} is tha Exclusive-OR of selected sources X end Y. The complemantar is programmed as a conditional inverter (AS11 = 1) dapendant on AS10. Tha X source is selected by inputs AS2 and AS3. The Y source is selected by AS0 and AS1 (AS4 = 1) end OR-ed with the accumulator (selected by $\overline{AS5}$·AS6). A selected logic function set is shown in Table 14.

Other functions end select line combinations ara possible with many redundant operations. Other conditions can be determined from previous truth tablas.

FIGURE 4 – BLOCK DIAGRAM OF ALU LOGIC OPERATION

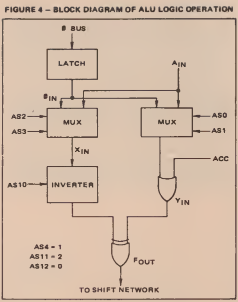
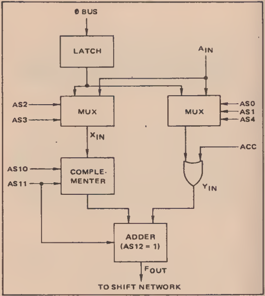


TABLE 14

Y MUX		X MUX		INV	ACC	FUNCTION
AS0	AS1	AS2	AS3	AS10	$\overline{AS5}$·AS6	
0	1	0	1	1	0	LOGIC 0
0	0	1	0	1	0	A
0	0	0	1	1	0	$\emptyset$
0	0	1	0	0	0	$\overline{A}$
0	0	0	1	0	0	$\overline{\emptyset}$
0	0	1	1	1	0	$A + \emptyset$
0	1	0	0	0	0	$A + \overline{\emptyset}$
1	0	0	0	0	0	$\overline{A} + \emptyset$
0	0	0	0	1	0	$A \cdot \emptyset$
0	1	1	1	1	0	$A \cdot \overline{e}$
0	1	0	0	1	0	$\overline{A} \cdot \emptyset$
0	1	1	0	1	0	$A \oplus \emptyset$
0	1	1	0	0	0	$\overline{A \oplus e}$
0	0	0	0	0	0	$\overline{A \cdot \emptyset}$
0	0	1	1	0	0	$\overline{A + \emptyset}$
0	1	0	1	0	0	LOGIC 1
1	0	1	0	1	1	$ACC \cdot \overline{A}$
0	1	0	1	1	1	$ACC \cdot \overline{\emptyset}$
1	0	1	0	0	1	$\overline{ACC} + A$
0	1	0	1	0	1	$\overline{ACC} + \emptyset$
0	0	1	0	1	1	$ACC \oplus A$
0	0	1	0	0	1	$ACC \oplus \overline{A}$
0	0	0	1	1	1	$ACC \oplus \emptyset$
0	0	0	1	0	1	$ACC \oplus \overline{\emptyset}$
0	0	0	0	1	1	$ACC \oplus A \cdot \emptyset$
0	0	0	0	0	1	$ACC \oplus \overline{A \cdot \emptyset}$
0	0	1	1	1	1	$ACC \oplus A + \emptyset$
0	0	1	1	0	1	$ACC \oplus \overline{A + \emptyset}$

+ = Logical Inclusive OR
· = Logical AND
⊕= Logical Exclusive OR

FIGURE 5 – BLOCK DIAGRAM OF ALU ARITHMETIC OPERATION



ALU ARITHMETIC OPERATION FUNCTION SET

The block diagram for arithmetic operation is similar to logic operation, however, the complementer and adder go to arithmetic mode. Select input AS12 is set to logic 1 for adder operation, however, AS4 is now used for increment/decrement by 2 and AS11 selects the binary or BCD operation.

The various arithmetic functions in the 4-Bit Slice are determined by the choice of operands to the adder. Most binary functions have a BCD equivalent, however, operands for BCD functions should be valid BCD characters.

Table 15 shows a selected arithmetic function set. Similar to the logic function set other combinations of select lines and operations are possible. These can be generated as needed by the previous truth tables.

TABLE 15

Y MUX		X MUX		±2	COMPLE-MENT	ACC	BINARY FUNCTION (PLUS C_{IN})	BCO FUNCTION (PLUS C_{IN})
AS0	AS1	AS2	AS3	AS4	AS10	$\overline{AS5} \cdot AS6$	AS11 = 1	AS11 = 0
1	0	0	1	1	1	0	A PLUS Ø	A PLUS Ø
1	0	0	1	1	0	0	A PLUS $\overline{\emptyset}$	A PLUS 9's COMP. Ø
0	1	1	0	1	0	0	Ø PLUS $\overline{A}$	Ø PLUS 9's COMP. A
0	0	1	0	1	1	0	A	A
0	0	0	1	1	1	0	Ø	Ø
0	0	1	0	1	0	0	$\overline{A}$	9's COMP. A
0	0	0	1	1	0	0	$\overline{\emptyset}$	9's COMP. Ø
1	1	1	0	1	1	0	−1 PLUS A	*
1	1	0	1	1	1	0	−1 PLUS Ø	*
1	1	1	0	0	1	0	−2 PLUS A	*
1	1	0	1	0	1	0	−2 PLUS Ø	*
0	0	1	0	0	1	0	+2 PLUS A	+2 PLUS A
0	0	0	1	0	1	0	+2 PLUS Ø	+2 PLUS Ø
1	0	1	0	1	1	0	A PLUS A	A PLUS A
0	1	0	1	1	1	0	Ø PLUS Ø	Ø PLUS Ø
0	0	1	0	1	1	1	ACC PLUS A	ACC PLUS A
0	0	0	1	1	1	1	ACC PLUS Ø	ACC PLUS Ø
0	0	1	0	1	0	1	ACC PLUS $\overline{A}$	ACC PLUS 9's COMP. A
0	0	0	1	1	0	1	ACC PLUS $\overline{\emptyset}$	ACC PLUS 9's COMP. Ø
0	0	0	0	1	1	1	ACC PLUS A·Ø	ACC PLUS A·Ø
0	0	0	0	1	0	1	ACC PLUS $\overline{A \cdot \emptyset}$	ACC PLUS 9's COMP. A·Ø
0	0	1	1	1	1	1	ACC PLUS A + Ø	*
0	0	1	1	1	0	1	ACC PLUS $\overline{A + \emptyset}$	*

*Not Defined in BCD

DATA ROUTING FUNCTION SET

Data routing in the MECL 4-Bit Slice covers the routing of data to and from both the shift network and accumulator. Data routing is controlled by select lines AS5, AS6, AS7, AS8, AS9, and AS15. AS5 and AS6 control the output destination of accumulator data, AS7 determines the source of data to the shift network, and AS8 enables and disables the input bus drivers. AS9 and AS15 control the source of data to both the accumulator and input bus drivers. Table 16 shows the truth table for AS7, AS8, AS9, and AS15.

The first four columns show all select line states. The fifth column shows the input source to the accumulator as controlled by AS9 and AS15. The possible accumulator inputs are: (1) ACC which is accumulator feedback on itself for accumulator clock disable. (2) IB which connects the input bus to the accumulator Inputs, (3) ØB which connects the output bus to the accumulator inputs, and (4) RES which connects the results of shift network to the accumulator input. The sixth column shows the two possible sources of input data to the shift network. These are from the accumulator (ACC) for accumulator shift operations, and from the ALU function outputs (F). The final column in the table shows the status of the input bus drivers. A logic 0 on AS8 disables the driver circuits so this part can be used to input data or for other system functions not related to the 4-Bit Slice. When enabled, the input bus port will output information from the accumulator or from the results of the shift network.

TABLE 16

AS7	AS8	AS9	AS15	FUNCTION		
				ACC SOURCE	SHIFT SOURCE	INPUT BUS
0	0	0	0	RES	ACC	DISABLE
0	0	0	1	ØB	ACC	DISABLE
0	0	1	0	IB	ACC	DISABLE
0	0	1	1	ACC	ACC	DISABLE
0	1	0	0	RES	ACC	ACC
0	1	0	1	ØB	ACC	RES
0	1	1	0	IB	ACC	RES
0	1	1	1	ACC	ACC	RES
1	0	0	0	RES	F_{DUT}	DISABLE
1	0	0	1	ØB	F_{OUT}	DISABLE
1	0	1	0	IB	F_{OUT}	DISABLE
1	0	1	1	ACC	F_{OUT}	DISABLE
1	1	0	0	RES	F_{OUT}	ACC
1	1	0	1	ØB	F_{OUT}	RES
1	1	1	0	IB	F_{OUT}	RES
1	1	1	1	ACC	F_{OUT}	RES

ELECTRICAL CHARACTERISTICS

Each MECL 10,000 series circuit hs been designed to meet the dc specifications shown in the test table, efter thermal equilibrium has been established. The circuit is in e test socket or mounted on a printed circuit boerd and trensverse eir flow greater then 500 linear fpm is maintained. Outputs ere terminated through a 50-ohm resistor to −2.0 volts. Test procedures are shown for only one input, or for one set of input conditions. Other inputs tested in the same manner.

RECOMMENDED OPERATING CONDITIONS – MC10800

PARAMETER	SYMBOL	VALUE	UNIT
Supply Voltage (V_{CC} = 0 Volts)	V_{TT} V_{EE}	−1.9 to −2.2 −4.68 to −5.72	Vdc Vdc
Operating Temp. (Functional)	T_A	−30 to +85	°C
Output Drive	–	50Ω to −2.0 Vdc	–
Maximum Clock Input Rise and Fall Time (20% to 80%)	t_r, t_f	10	ns
Minimum Clock Pulse Width	PW	5	ns

@Test Temperature	TEST VOLTAGE VALUES					
	Volts					
	V_{IHmax}	V_{ILmin}	V_{IHAmin}	V_{ILAmax}	V_{EE}	V_{TT}
30°C	−0.890	1 890	1 205	1 500	5 2	−2 0
+25°C	−0.810	−1.850	1 105	1 475	5 2	−2.0
+85°C	−0 700	1 825	1 035	1 440	5 2	−2.0

Characteristic	Symbol	Pin Under Test	MC10800 TEST LIMITS								VOLTAGE APPLIEO TO PINS LISTEO BELOW						(V_{CC}) Gnd
			−30°C		+25°C			+85°C									
			Min	Max	Min	Typ	Max	Min	Max	Unit	V_{IHmax}	V_{ILmin}	V_{IHAmin}	V_{ILAmax}	V_{EE}	V_{TT}	
Power Supply Orain Current	I_{EE} I_{TT}	1, 24 25 48		 –		195 180	240 199	– –		mAdc	– –	– –	– –	– –	1 24	25 48	12 36 7 17
Input Current	I_{inH} I_{inL}	23 31 27 31		- –	 0 5	 -	65 350 435	 –		µAdc µAdc	23 31 27 –	– – – 31	– – – –	– – – –	↓	↓	↓
Logic '0 Output Voltage	V_{OH}	13 10	1 060 1 060	−0.890 −0.690	−0 960 −0.960		−0.810 −0.810	−0.690 −0.690	−0.700 −0 700	Vdc Vdc	18 27 30,38,* 18.27.30.38,*	– –	– –	– –	↓	↓	↓
Logic ' 1 ' ** Output Voltage	V_{OL}	13 10	−1.940 −1.890	1 675 1 675	−1.900 −1.850		−1.650 −1.650	1 875 - 1 825	1 615 1 615	Vdc Vdc	30* *	– –	– –	– –	↓	↓	↓
Logic '0' Threshold Voltage	V_{OHA}	13 10	−1.080 −1.080		−0.980 −0.980			−0.910 −0.910		Vdc Vdc	27 29,** 29,**	– –	– –	18 47	↓	↓	↓
Logic "1" Threshold Voltage	V_{OLA}	13 10		1 655 1 655	 -		−1 630 −1.630		1 595 1 595	Vdc Vdc	44.46,** 29,**	– –	29 47	– –	↓	↓	↓

*V_{IH} on pins 19. 26. 29. 31. 32. 33. 34. 35. 37. 44. 46

**V_{IH} on pins 19. 26. 30. 31. 32. 33. 34. 35, 37

*** The bi directional outputs are specified at - 1 90 volts for V_{OL} min

SETUP AND HOLD TIMES (NANOSECONDS OVER TEMPERATURE RANGE).

Input	Path	Mode	Setup Max	Hold Max
A Bus	→ A MUX → MASK MUX → COMP → ALU → SHIFT → ACC	Arith Subtract	38.0	−15.0
A 8us	→ A MUX → Y MUX → ALU → SHIFT → ACC	Logical	19.0	−5.0
ϕ Bus	→ LATCH → MASK MUX → COMP → ALU → SHIFT → ACC	Arith Subtract	38.0	−15.0
ϕ Bus	→ LATCH → Y MUX → ALU → SHIFT → ACC	Logical	20.0	−5.0
ϕ Bus	→ ACCUMULATOR	Direct	7.0	+5.0
I Bus	→ ACCUMULATOR	Direct	7.0	+5.0
AS0, 1	Y MUX → ALU → SHIFT → ACC	Arith Add	32.0	−15.0
AS4	ALU → SHIFT → ACC	Logical	15.0	0.0
AS3	MASK MUX → COMP → ALU → SHIFT → ACC	Arith Subtract	36.0	−17.0
AS2	MASK MUX → ALU → SHIFT → ACC	Logical (No Comp)	22.0	−5.0
AS5, 6	ALU → SHIFT → ACC		20.0	−5.0
AS7	SHIFT INPUT MUX → SHIFT → ACCUMULATOR	Direct	10.0	+5.0
AS9, 15	ACCUMULATOR INPUT MUX → ACC	Oirect	8.0	+7.0
AS10	COMP → ALU → SHIFT → ACC	Arith	35.0	−15.0
AS11	ALU → SHIFT → ACC	Arith	21.0	−2.0
AS12	ALU → SHIFT → ACC		28.0	−10.0
AS12	ALU → SHIFT → ACC		14.0	+2.0
AS13, 14	SHIFT NETWORK → ACCUMULATOR	Direct	16.0	+5.0
CIN	→ ALU → SHIFT → ACC	Arith	19.0	+2.0
R-1, R4	SHIFT NETWORK → ACCUMULATOR	Direct	8.0	+5.0
ϕ Bus	→ LATCH (AS16 – LATCH CLOCK)	Oirect	5.0	+6.0

PROPAGATION DELAYS (NANOSECONDS)

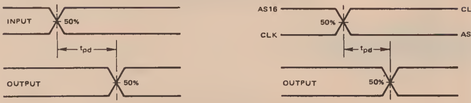
Input	Path			Output	-30°C T_A		+25°C T_A		+85°C T_A	
	Via	Moda	Function		Typ	Max	Typ	Max	Typ	Max
A Bus ØBus	ALU	Arith	Subtract	I Bus	30.0	39.0	32.0	41.0	37.0	49.0
				PG, GG	16.0	21.0	17.5	21.0	20.0	27.0
				C_{OUT}	18.0	22.0	19.0	23.0	22.0	28.0
				OF, ZD R-1, R4	27.0	37.0	29.5	39.0	34.0	44.0
				PC, PR	27.0	34.0	29.0	36.0	34.0	41.0
C_{IN}	ALU	Arith	Addition	I Bus	15.0	18.5	16.0	19.5	19.0	24.5
				C_{OUT}	5.0	7.0	5.5	7.5	6.0	8.5
				OF, ZD R-1, R4	12.5	16.0	13.5	17.0	15.5	19.0
				PC, PR	13.5	18.0	14.5	19.0	17.0	23.0
AS0 AS1 AS2 AS3 AS4 AS5 AS6 AS10 AS11 AS12	ALU	Arith	Subtract Accumulator	I Bus	36.0	43.0	38.5	46.5	47.0	64.0
				PG, GG	23.0	30.0	24.0	30.0	30.0	38.0
				C_{OUT}	24.0	32.0	26.0	32.0	31.5	39.0
				OF, ZD R-1, R4	33.0	43.0	36.0	46.0	47.0	60.0
				PC, PR	33.0	40.0	35.0	42.0	44.0	57.0
AS16	ALU	Arith	Subtract	I Bus	33.0	40.0	35.0	43.0	41.0	51.0
				PG, GG	20.0	25.0	21.0	26.0	25.0	32.0
				C_{OUT}	21.5	26.0	23.0	27.5	26.5	33.0
				OF, 2D R-1, R4	30.5	39.0	33.0	42.0	38.0	47.0
				PC, PR	30.5	36.0	33.0	39.0	38.0	47.0
R-1 R4	Shift	Shift Left Shift Right	–	I Bus	7.0	8.5	7.5	9.0	9.0	13.0
AS7 AS13 AS14	Shift	Shift Left Shift Right	–	I Bus	10.0	16.0	10.0	16.0	12.5	18.0
AS9 AS15	Direct	Shift ACC	–	I Bus	8.0	11.0	8.5	11.5	10.0	13.5
AS8	Direct	Enable Disable	–	I Bus	5.5	8.5	6.0	8.5	7.5	10.0
AS5 AS6	Direct	Enable Disable	–	Ø Bus	7.0	9.5	7.5	9.5	10.0	17.0
CLK	A Bus ALU	Arith	Subtract Accumulator	I Bus	38.5	48.0	41.0	51.0	47.0	57.0
				PG, GG	26.0	36.0	27.5	38.0	31.0	43.0
				C_{OUT}	27.5	38.0	29.0	40.0	32.5	45.0
				OF, ZD R-1, R4	37.0	41.0	39.0	43.0	44.5	49.0
				PC, PR	36.5	44.0	39.0	46.0	44.0	55.0
CLK	ALU	Arith	Add Accumulator	I Bus	34.5	45.0	36.5	47.0	42.5	58.0
CLK	Shift	AS7 = 0	Multiple Shift	I Bus	13.0	17.5	14.0	18.5	16.0	21.0
				OF, ZD R-1, R4	14.0	18.0	14.5	19.0	16.0	23.0
				PC, PR	15.0	20.0	16.0	21.0	18.0	24.0
CLK	Direct	–	Acc to I Bus	I Bus	8.0	11.0	8.5	11.0	10.0	13.0

PROPAGATION DELAYS (NANOSECONDS) (continued)

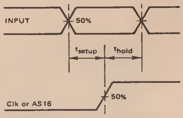
Input	Path			Output	-30°C T_A		+25°C T_A		+85°C T_A	
	Via	Mode	Function		Typ	Max	Typ	Max	Typ	Max
CLK	Direct	–	Acc to ØBus	ØBus	B.5	12.0	9.0	12.0	10.0	13.0
ØBus	ALU (Mask)	Logic	Without Complement	I Bus	23.0	32.0	25.0	35.0	30.0	45.0
CLK	A Bus (Mask)	Logic	Without Complement	I Bus	33.0	42.0	35.0	43.0	40.0	47.0
CLK	A Bus (Mask)	Logic	With Complement	I Bus	34.5	44.0	37.0	47.0	42.0	55.0
CLK	A Bus (Y Mux)	Logic	Without Complement	I Bus	31 0	39.0	33.0	41.0	37.0	44.0
Output Rise and Fall Time (20% – B0%)				All	3.0	5.0	3.5	5.5	4.0	6.0

SWITCHING WAVEFORMS

PROPAGATION DELAYS



SETUP AND HOLD



TEST PROCEDURE:

e) Establish setup time with long t_{hold}.

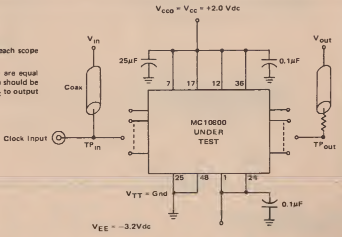
b) Keeping the leading edge of the input constent (t_{setup}) vary the trailing edge of the input to determine t_{hold}.

NOTE: t_{setup} and t_{hold} as defined are positive. Internal delays in the date path may result in a shift of the dete waveform to tha left, with respect to the clock, resulting in negative hold times.

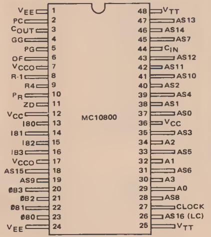
SWITCHING TIME TEST CIRCUIT

50 ohm termination to ground located in each scope channal input.

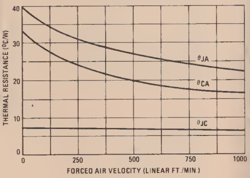
All input and output cables to the scope are equal lengths of 50 ohm coaxial cable. Wire length should be <¼ inch from TP_{in} to input pin and TP_{out} to output pin.



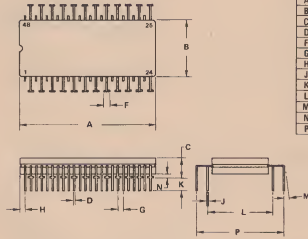
PIN ASSIGNMENT



THERMAL CHARACTERISTICS (TYPICAL)



PACKAGE DIMENSIONS



	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	31.24	32.26	1.230	1.270
B	12.70	13.72	0.500	0.540
C	4.57	5.59	0.180	0.220
D	0.38	0.53	0.015	0.021
F	1.14	1.40	0.045	0.055
G	1.27 BSC		0.050 BSC	
H	1.02	1.52	0.040	0.060
J	0.20	0.30	0.008	0.012
K	2.54	3.30	0.100	0.130
L	15.24 8SC		0.600 BSC	
M	–	7°	–	7°
N	0.51	1.52	0.020	0.060
P	20.32 8SC		0.800 8SC	

Case 725–01

A socket for the QUIL package is available from ELECTRONIC MOLDING CORPORATION. (Part number 7178-295-5)

QUIL is a trademark of Motorola Inc.

MOTOROLA

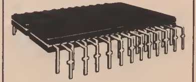
MC10801

INTRODUCTION

The MC10801 Microprogram Control Function is an LSI building block for digital processor systems. This circuit controls machine operations by generating the addresses and sequencing pattern for microprogram control storage. The MC10801 is compatible with a wide range of control memory sizes and organizations. Each part is 4 bits wide and can be connected in parallel for larger memory addresses. Maximum system flexibility is maintained with 5 separate data ports.

The Microprogram Control Function as shown in the block diagram below contains a control memory address register CRO, multi-purpose registers CR1-CR3, an incrementer, a subroutine LIFO, and the associated next address, status, and bus control logic in a single MECL Bipolar LSI circuit. Nine select (CS) lines and four instruction inputs (IC̄) control all operations within the part.

MECL — LSI MICROPROGRAM CONTROL FUNCTION



CASE 725-01

Microprogram Control Function BLOCK DIAGRAM—MC10801

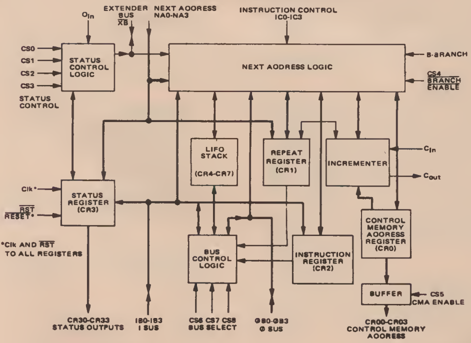
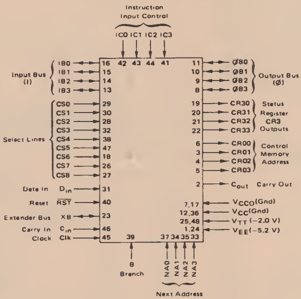


TABLE OF CONTENTS

IMPORTANT FEATURES

1. 16 microprogram sequencing instructions including:
   - a. Increment
   - b. Direct jumps
   - c. Conditional jumps
   - d. Subroutining
   - e. Conditional Subrouting
2. 4-bit registers expandable with parallel MC10801 circuits.
   - a. Microprogram address register – CRO
   - b. Repeat register – CR1
   - c. Instruction register – CR2
   - d. Status register – CR3
3. Expandable 4 X 4 push-pop stack for nesting subroutine – CR4-CR7.
4. Branch inputs for conditional operations and multiway branching
5. Address masking on special instructions.
6. Repeat logic for repeating subroutines or single instructions.
7. All registers are of edge triggered master-slave design.
8. Fully compatible with the MECL 10,000 family.

INPUT/OUTPUT DIAGRAM–MC10801



ABSOLUTE MAXIMUM RATINGS (see Note 1)

RATING		SYMBOL	VALUE	UNIT
Supply Voltage (V_{CC} = 0)		V_{EE} V_{TT}	−8 to 0 −4 to 0	Vdc Vdc
Input Voltege (V_{CC} = 0)	Std Bus	V_{in} V_{in}	0 to V_{EE} Note 2	Vdc Vdc
Output Source Current	Cont Surge	I_o I_o	< 50 <100	mAdc mAdc
Storage Temp. Junction Temp.		$T_{stg.}$ T_J	−55 to +150 165	°C °C

NOTE: 1. Permanent device demage may occur if absolute maximum retings ere exceeded. Functionel operetion should be restricted to RECOMMENDED OPERATING CONDITIONS. Exposure to higher then recommended voltages for extended periods of time could affect device reliability.

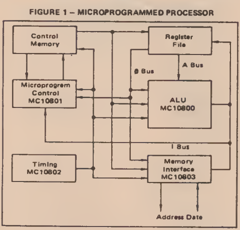
NOTE: 2. Input voltage limit is V_{CC} to −2 Volts when the bus is used as en input end the output drivers ere disabled.

SYSTEM OVERVIEW

The Motorola M10800 family of LSI processor circuits has been partitioned into key building block elements as shown in Figure 1. The LSI circuits can be interconnected and programmed for a wide range of processor system applications. Combinations of the various circuits allow expansion to any required data word length or control memory size. Multiple I/O ports on each circuit provide maximum data flow flexibility. The M10800 LSI family is designed to provide functional systam blocks without limiting a final system siza or architecture.

The M10800 system is designed around a microprogrammed concept for greatest versatility. Microprogramming permits emulating existing machines or software, updating systems by adding more capability, or modifying systems to meet specific customer requirements. The microprogram is contained in the control memory block of Figure 1. Depending on systam requiraments, this memory can vary from a few hundred words to savaral thousand. Tha size and organization of this block is controlled by tha system dasigner and is constructed with MECL PROMs such as tha MCM10149, or MECL RAMs such as the MCM10144 or MCM10146.

FIGURE 1 – MICROPROGRAMMED PROCESSOR



In a microprogrammed processor the information for executing a machine function (macroinstruction) is contained within the control memory. Control memory outputs go to the Register File, ALU, and Memory Interface blocks in Figura 1 and control the specific function performed by aach section of the processor. The number of control mamory staps (microinstructions) required to axecute a macroinstruction is datermined by complexity of the instruction. For example, a simple registar to register add can require only ona or two microinstructions, while a more complex multiply or floating point arithmetic calculation requires several control memory words addressed in tha proper sequence.

The heart of a microprogrammed systam is tha microprogram control logic. This block in Figure 1 holds tha present control mamory word address and controls tha sequencing to execute processor operations. Within the Motorola M10800 family, the MC10801 Microprogram Control Function performs this important task. Each circuit is four bits wide and parallel combinations adapt to any required control memory size. A set of sixteen instructions address the MC10801 and control the sequencing of the microprogram storage. Powerful branch and subroutine instructions incraase system performanca and minimize tha amount of control memory required to build a system. The sixtaan instructions ease the burden of writing a microprogram by expressing program flow in a manner familiar to assembly language programmars.

Versatility is a key word to describe each circuit in tha Motorola M10800 family. Tha block diagram in Figura 1 and the axamples in this data sheet ara intended to illustrate ways to use these LSI parts and do not restrict the designer to any particular system configuration or application.

PIN ASSIGNMENTS

Pin Designation	Pin Number	Description
IC0	42	Instruction Control Input
IC1	43	Instruction Control Input
IC2	44	Instruction Control Input
IC3	41	Instruction Control Input
IB0	16	Input Bus – LSB I/O
IB1	15	Input Bus – NLSB I/O
IB2	14	Input Bus – NMSB I/O
IB3	13	Input Bus – MSB I/O
ØB0	11	Output Bus – LSB I/O
ØB1	10	Output Bus – NLSB I/O
ØB2	9	Output Bus – NMSB I/O
ØB3	B	Output Bus – MSB I/O
NA0	37	Next Address – LSB Input
NA1	34	Next Address – NLSB Input
NA2	35	Next Address – NMSB Input
NA3	33	Next Address – MSB Input
CR00	6	Control Memory Address – LSB Output
CR01	3	Control Memory Address – NLSB Output
CR02	4	Control Memory Address – NMSB Output
CR03	5	Control Memory Address – MSB Output
CR30	19	Status Register CR3 Output
CR31	20	Status Register CR3 Output
CR32	21	Status Register CR3 Output
CR33	22	Status Register CR3 Output
CS0	29	Status Register Control – Select Input
CS1	30	Status Register Control – Select Input
CS2	2B	Status Register Control – Select Input
CS3	32	Status Register Control – Select Input
CS4	3B	Branch Line – Select Input
CS5	47	Control Memory Address – Enable Input
CS6	1B	Ø Bus/I Bus Control – Select Input
CS7	26	Ø Bus/I Bus Control – Select Input
CSB	27	Ø Bus/I Bus Control – Select Input
C_{in}	46	Carry Input
C_{out}	2	Carry Output
D_{in}	31	Data Input to CR3
B	39	Branch Input
$\overline{XB}$	23	Extender Bus
$\overline{RST}$	40	Reset Input
Clk	45	Clock Input
V_{EE}	1	–5.2 Volt Supply
V_{EE}	24	–5.2 Volt Supply
V_{TT}	25	–2.0 Volt Supply
V_{TT}	48	–2.0 Volt Supply
V_{CC}	12	Ground
V_{CC}	36	Ground
V_{CCO}	7	Ground
V_{CCO}	17	Ground

ARCHITECTURAL DESCRIPTION

The MC10801 Microprogram Control Function is composed of B master slave registers, CR0 through CR7, as shown in Figure 2. Additional gates, multiplexers, and a next address logic block transfer information to and from these registers. Five 4-bit data ports (CR0, CR3, NA, I Bus, and Ø Bus) are available to enter and output address information. In addition, three single line terminals (B, $\overline{\text{XB}}$, and D_{in}) provide status inputs for decisions within the part. Each of the eight registers fills an important function in the storage and generation of control memory addresses. The individual registers and data transfer paths in Figure 2 are described below.

CR0 — CONTROL MEMORY ADDRESS REGISTER

Register CR0 holds the present microprogram control memory address and its outputs are gated to package pins CR00 through CR03. In a system these outputs address the control memory storage block. The next address logic block in Figure 2 generates next address information to the CR0 register inputs. A positive clock edge loads the new control memory address into CR0 which in turn selects the next microinstruction.

NEXT ADDRESS LOGIC

The next address logic block performs 16 sequence instructions as selected by the instruction control lines IC0 through IC3 inputs. These 16 control instructions, see Table 1, determine the source of control memory address information within each MC10801. Possible sources are CR1, CR2, CR4, NA inputs, I Bus, Ø Bus, and the incrementer. During each microcycle the next address block generates a new control memory address in parallel with other processor functions, such as the ALU. Detailed information on the 16 MC10801 instructions follows in the Functional Description section of this data sheet.

FIGURE 2 — FUNCTIONAL BLOCK DIAGRAM

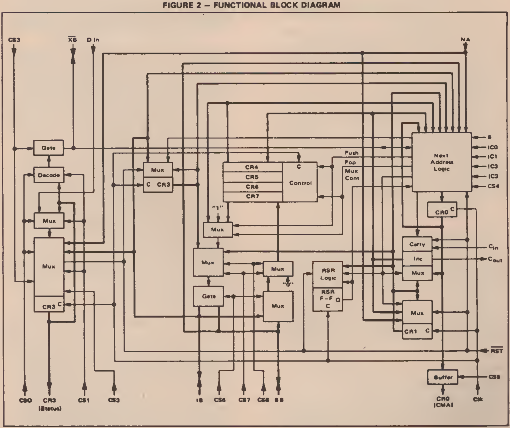


TABLE 1
MC10801 CONTROL INSTRUCTIONS

INC	–	Increment
JMP	–	Jump to N.A. Inputs
JIB	–	Jump to I Bus
JIN	–	Jump to I Bus and Load CR2
JPI	–	Jump to Primary Instruction (CR2)
JEP	–	Jump to External Port (Ø Bus)
JL2	–	Jump to N.A. Inputs and Load CR2
JLA	–	Jump to N.A. Inputs and Load Address into CR1
JSR	–	Jump to Subroutine
RTN	–	Return from Subroutine
RSR	–	Repeat Subroutine (Load CR1 from N.A. Inputs)
RPI	–	Repeat Instruction
BRC	–	Branch to N.A. Inputs on Condition; otherwise Increment
BSR	–	Branch to Subroutine on Condition; otherwise Increment
ROC	–	Return from Subroutine on Condition; otherwise Jump to N.A. Inputs
BRM	–	Branch and Modify Address with Branch Inputs (Multiway Branch)

CR1 – REPEAT REGISTER

Register CR1 is primarily designed to be an index counter for repeating single microinstructions or repeating subroutines. This repeat feature is important for multiple shift, multiply, and divide machine instructions. To perform a microprogram repeat sequence, the repeat count is first loaded into CR1 from the NA inputs with a RSR-Repeat Subroutine instruction (Table 1). Each time the selected microinstruction or subroutine is executed, CR1 is automatically incremented. Upon reaching the final repeat count the MC10801 continues to the next microprogram instruction.

A second function performed by CR1 is a control memory address save register. In this mode the present control memory address in CR0 is transferred to CR1 on a JLA-Jump and Load Address instruction (Table 1). At a later time it is possible to return to the stored address by transferring CR1 back to CR0 on a RPI-Repeat Instruction command.

The operation of CR1 is controlled by the next address logic. Possible input sources are the NA inputs, the incrementer, and CR0. CR1 outputs are routed to either CR0, to the incrementer, or to a Bus output.

CR2 – INSTRUCTION REGISTER

Register CR2 is used primarily as an instruction or op code storage register. After fetching a machine instruction, the control memory starting address can be stored in CR2. It can then be used later by transferring the contents of CR2 through the next address logic to the control memory address register CR0. As with register CR1, the operation of CR2 is controlled by the instruction inputs IC0 – IC3 and the next address logic. The I Bus is the source for CR2 and is loaded on either a JIN or JL2 instruction (Table 1). Information is transferred from CR2 to CR0 on a JPI-Jump to Primary instruction.

CR2 is not limited to an instruction register and can be used anytime it is desirable to store a control memory address location for future use. For example, CR2 can store an interrupt vector which may be loaded into CR0 as needed.

CR3 – STATUS REGISTER

Register CR3 is normally used as a status register for storing flag conditions. This 4-bit register can be parallel loaded from either the NA or I Bus inputs. In additicn, any single CR3 bit can be set or cleared from the D_{in} input. The CR3 outputs are continuously available on the CR30 to CR33 package pins. The CR3 status information may be used in conjunction with other external information for generating branch conditions.

Any single CR3 bit can be selected and gated onto the $\overline{XB}$ extender bus line. $\overline{XB}$ goes to the next address logic to control branch decisions. When MC10801's are operated in parallel, the $\overline{XB}$ line is common to every part. Therefore, branch decisions can be made independent of which MC10801 circuit contains the selected status bit. The operation of CR3 with respect to the I Bus, NA inputs, D_{in}, and $\overline{XB}$ is controlled by select lines CS0, CS1, CS2, and CS3.

Another use for CR3 is to extend the control memory address. This is accomplished by organizing the control memory in a word-page format. The word address is contained in CR0 and the page address in CR3. With two MC10B01's each page can be 256 words (B CR0 bits) and 16 pages may be addressed with 4 CR3 bits or 256 possible pages using all B CR3 bits.

A third use for CR3 is to store all or part of the instruction operation code. In this manner, individual op code bits could be selected onto the $\overline{XB}$ line and tested for secondary decode decisions.

CR4 – CR7 LIFO STACK

Registers CR4 through CR7 are connected as a last-in-first-out (LIFO) stack for nesting subroutines within microprogram. When jumping to a subroutine, the return destination is automatically pushed onto the top of the LIFO (CR4). When returning from subroutine, CR4 is loaded into the control memory address register CR0.

With 4 registers it is possible to nest subroutines up to 4 deep within the LIFO. If additional stack depth is required CR1 can be used as a fifth location or CR7 can be expanded to any length through the I Bus or O Bus ports to additional MECL MSI circuits.

Reading CR7 via the I Bus or Ø Bus during a push of the LIFO stack provides a means for testing when the stack is full. Logic "0" bits are normally stuffed into the bottom of the stack on a "pop" or read operation. Therefore, any information in CR7 would indicate the stack is full.

Push and pop stack operations are controlled by the IC0 — IC3 inputs and the next address logic. In addition, select lines CS6, CS7, and CS8 route information to and from the LIFO via the I Bus and Ø Bus ports.

INCREMENTER

The 4-bit incrementer is used in several of the Table 1 microprogram control instructions. One is the INC-Increment command which linearly steps through a microprogram. A second function is to increment CR1 when it is used as an index counter for repeating microinstructions or subroutines as described in the earlier CR1 section. Increment is also used with the JSR-Jump to Subroutine, BSR-Branch to Subroutine, and JLA-Jump and Load Address commands to generate the proper return address. Operation of the incrementer is controlled by the IC0 — IC3 code and the C_{in} input.

The incrementer is expanded with the carry in (C_{in}) and carry out (C_{out}) terminals when MC10801 circuits are operated in parallel. The carry out of one MC10801 is connected directly to carry in of the circuit handling the next most significant control memory address bits. Carry out of the most significant bit is not required for count operation, but it can be used to signify maximum count value at the incrementer inputs.

Carry in to the least significant MC10801 is connected to a logic "1" for the increment operation. This input is normally hard wired, but in some applications can be system controlled to override the incrementer.

RSR LOGIC AND RSR FLIP FLOP

The repeat subroutine (RSR) logic and flip flop blocks in Figure 2 provide a means for setting the MC10801 in an instruction repeat sequence as described in the previous CR1 section. The RSR flip flop is automatically set when a repeat constant is loaded into CR1 with a RSR-Repeat Subroutine instruction. It is cleared when CR1 reaches the final repeat count. By monitoring the RSR flip flop status, the MC10801 can decide when to perform microinstruction repeats. Additional details of the RSR flip flop operation are explained in the following Functional Description section.

CLK — CLOCK

All registers in the MC10801 Microprogram Control Function are composed of master-slave flip flops and must be clocked to change stored data. A common clock is routed directly to all eight registers. As is characteristic of MECL flip flops, the registers are clocked on the positive going (V_{OL} to V_{OH}) clock edge. At that time data present on the register inputs is stored in the register and is available at the register outputs. Signals on the register inputs can change at any time, with the clock input at either logic state, and not change the register outputs. The only restriction on changing register inputs is during the set up and hold time near the positive going clock edge.

$\overline{RST}$ — RESET

The $\overline{RST}$ input is held at MECL V_{OL} during normal system operation. However, by forcing this input to the MECL V_{OH} level, it is possible to reset all registers in the MC10801. Reset operates in conjunction with the clock and therefore is a synchronous reset. Reset is accomplished in the following sequence. CR0, CR1, CR2, and CR3 are reset on the first clock pulse. The LIFO is connected to the incrementer to which carry-in is inhibited. The LIFO is also forced to a push mode during reset. Therefore, a maximum of five clock pulses reset all MC10801 registers in the following sequence: CR0/CR1/CR2/CR3, CR4, CR5, CR6, CR7.

FUNCTIONAL DESCRIPTION

MICROPROGRAM SEQUENCE CONTROL INSTRUCTIONS IC0-IC3

The MC10801 generates the microprogram address sequencing from 16 control instructions which are encoded on the IC0 — IC3 inputs. Each control instruction determines the data source for the next microprogram control memory address. This next address information is then stored in register CR0 on a positive going clock signal.

The 16 sequence control instructions are each described in Table 2. Table 2 lists these instructions and shows the associated mnemonics, binary select codes, and register transfers. Several instructions require making decisions on the status of the branch (B), extender bus ($\overline{XB}$), RSR flip flop output (RSQ), or select line CS4. Both decision alternatives are given for these instructions.

INC — Increment

The increment command routes the present contents of CR0 through the incrementer, adds C_{in}, and multiplexes the result (CR0 plus C_{in}) to the CR0 register inputs. As with all control instructions, the new address is loaded on a positive clock transition. This instruction is used to linearly step through the microprogram memory. When MC10801s are operated in parallel, C_{in} of a more significant device is connected to C_{out} of the previous MC10801. The least significant C_{in} is normally left floating at a logic "1".

JMP — Jump to Next Address

The JMP command provides for an unconditional jump to another control memory address. The jump destination is directly supplied on the NA inputs, which are normally feedback from control memory. A clock transition transfers address data from the NA inputs to register CR0.

FUNCTIONAL DESCRIPTION

Four instruction control inputs, IC0 – IC3, and nine select lines, CS0 – CS8, control the flow of data within the MC10801 Microprogram Control Function. The following information describes programming these inputs to perform the various circuit functions. *All truth tables are expressed in negative logic with V_{OL} being a logic 1 and V_{OH} a logic 0.*

TABLE 2[5]

	CODE							REGISTER AND FLIP FLOP OUTPUTS[4] V_{OL} ⟋ V_{OH}				
MNEM	IC3	IC2	IC1	IC0	DESCRIPTION	RESET $\overline{RST}$	BRANCH OR REPEAT CONDITION[2]	CR0[7]	CR1	CR2	LIFO STACK CR4 – CR7[6]	RSQ[3]
X	X	X	X	X	RESET CONDITION	0	X	0	0	0	"PUSH" CR0 TD STACK	0
INC	1	1	0	0	INCREMENT	1	X	CR0 plus C_{in}	–	–	–	–
JMP	0	0	1	0	JUMP TD NEXT ADDRESS	1	X	NA	–	–	–	–
JIB	1	0	0	0	JUMP TO I BUS	1	X	IB·NA	–	–	–	–
JIN	1	0	0	1	JUMP TO I BUS & LOAD CR2	1	X	IB·NA	–	IB	–	–
JPI	1	0	1	0	JUMP TO PRIMARY INST	1	X	CR2·NA	–	–	–	–
JEP	1	1	1	0	JUMP TO EXTERNAL PORT	1	X	ØB·NA	–	–	–	–
JL2	0	0	0	1	JUMP & LOAD CR2	1	X	NA	–	IB	–	–
JLA	0	0	1	1	JUMP & LDAD ADDRESS	1	X	NA	CR0 plus C_{in}	–	–	–
JSR	0	0	0	0	JUMP TO SUBROUTINE	1	$\overline{RSQ}$+RIN·$\overline{XB}$=0	NA	–	–	"PUSH" CR0 TO STACK	–
						1	$\overline{RSQ}$+RIN·$\overline{XB}$=1	NA	–	–	"PUSH" CR0 plus C_{in}	–
RTN	1	1	1	1	RETURN FROM SUBROUTINE	1	$\overline{RSQ}$+RIN·$\overline{XB}$=0	CR4	CR1 plus C_{in}	–	"POP" STACK TO CR0	–
						1	$\overline{RSQ}$+RIN·$\overline{XB}$=1	CR4	–	–	"POP" STACK TO CR0	0
RSR	1	1	0	1	REPEAT SUBROUTINE	1	X	CR0 plus C_{in}	NA	–	–	1
RPI	1	0	1	1	REPEAT INSTRUCTION	1	$\overline{RSQ}$+RIN·$\overline{XB}$=0	–	CR1 plus C_{in}	–	–	–
						1	$\overline{RSQ}$+RIN·$\overline{XB}$=1	CR1·NA	–	–	–	0
BRC	0	1	0	1	BRANCH ON CONDITION	1	$\overline{XB}$·(CS4+$\overline{B}$)=0	NA	–	–	–	–
						1	$\overline{XB}$·(CS4+$\overline{B}$)=1	CR0 plus C_{in}	–	–	–	–
BSR	0	1	0	0	BRANCH TO SUBROUTINE	1	$\overline{XB}$·(CS4+$\overline{B}$)=0	NA	–	–	"PUSH" CR0 plus C_{in}	–
						1	$\overline{XB}$·(CS4+$\overline{B}$)=1	CR0 plus C_{in}	–	–	–	–
ROC	0	1	1	1	RETURN DN CONOITION	1	$\overline{XB}$·(CS4+$\overline{B}$)=0	CR4	–	–	"POP" STACK TO CR0	–
						1	$\overline{XB}$·(CS4+$\overline{B}$)=1	NA	–	–	–	–
BRM	0	1	1	0	BRANCH & MODIFY	1	CS4=1	NA	–	–	–	–
						1	CS4=0	CR00=NA0·B CR01=NA1·XB CR02=NA2 CR03=NA3	–	–	–	–

NOTES.

1. X = DON'T CARE STATE
   – = NO CHANGE
2. EQUATIONS APPLY AS SHOWN, WHERE:
   RIN = (CR13·CR12·CR11·CR10)
   $\overline{XB}$ = EXTERNAL EXTENDER BUS NDDE (see Table 3)
   $\overline{B}$ = COMPLEMENT OF BRANCH INPUT
3. RSD = OUTPUT OF RSR FLIP FLOP
4. ALL REGISTERS AND RSR FLIP FLOP CHANGE STATE ON V_{OL} TO V_{OH} (POSITIVE GDING) CLOCK TRANSITION
5. NEGATIVE LOGIC USED THROUGHDUT
6. TABLE 8 SHOWS LIFO STACK TRUTH TABLE
7. CR0 CHIP OUTPUTS ENABLED WHEN CS5=1

JIB — Jump to I Bus

The JIB instruction is a direct jump to address information on the I Bus port. Tha I Bus is normally an internal data bus in tha processor and can be used to input the starting address of a microprogram instruction routina. The I Bus data is modified or "masked" with control memory feedback on the NA inputs. The next address is therefore determined by I Bus ANDed with NA inputs.

JIN — Jump to I Bus and Load CR2

The JIN command routes the I Bus ANDed with NA input to CR0 as does JIB. In addition, JIN loads unmodified I Bus data into register CR2 on the same clock edge. This information in CR2 can be used at a later point in microprogram for primary and secondary program flow modification.

JPI — Jump to Primary Instruction

The JPI command is a jump to the contents of CR2 ANDed with the NA inputs. Register CR2 is loaded with a previous JIN or JL2 instruction. The code stored in CR2 is used to start a new sequance of microinstructions or modify tha present microinstruction sequence.

JEP — Jump to External Port

The JEP instruction is a direct jump to information on the Ø Bus port. Tha Ø Bus data is ANDed with the NA inputs (Ø Bus·NA) prior to entaring register CR0. This instruction offers an additional port to enter a starting address or modify information to microprogram flow.

JL2 — Jump to NA Inputs and Load CR2

Tha JL2 command is a direct jump to tha NA inputs and a parallel load of CR2 from tha I Bus. This instruction allows CR2 to be loeded during the axecution of another microinstruction. This is useful for storing an interrupt vector or a new operation address while finishing a previous microinstruction sequence.

JLA — Jump to NA Inputs and Load CR1

Tha JLA command is a direct jump to tha NA inputs and a parallal load of CR1. CR1 is loeded with the incremanted value of CR0 (CR0 plus C_{in}). The JLA instruction can be used to service an interrupt or as an additional form of subroutining.

JSR — Jump to Subroutine

The JSR instruction is an unconditional jump to subroutine. The jump address is provided by the NA inputs which are loaded into register CR0. At the same time, the present CR0 address is routed through the incrementer and "pushed" onto tha LIFO stack to CR4.

The JSR command operates in two modes depending upon the status of the RSR flip flop (see Table 2).

1. Non-Repeat mode is used for normal subroutining. The RSR flip flop is clear (RSQ = 0) which causes the present CR0 address to be incremented and pushed onto the stack. That is, CR0 plus $C_{in} \rightarrow$ CR4 and the contents in registers CR4 through CR7 are "pushed down" one location. Upon a return from subroutine, the incremented address puts the control into the main program flow one location below the JSR address.

2. Repeat mode is used for multiple executions of e single subroutine. The RSR flip-flop has been previously set (RSQ = 1) by en RSR instruction.

   Tha incramanter is disabled and CR0 is loaded into CR4. The stack registers CR4—CR7 are pushed down as befora. Upon a return from subroutina, tha original JSR address is then returned to CR0 and tha JSR is executed again. This repeat cycle will continue until $\overline{XB}$ signifies the final repeet count has been reached.

   For multipla MC10801 configurations, the $\overline{XB}$ line is a common connection between parallel circuits. Tha RSR flip-flop significes the repeat mode and $\overline{XB}$ combines with CR1 ragisters to determine tha final repeat count. During a JSR instruction tha incramantar is controlled by the following equation:

   $$\text{INTERNAL CARRY\_IN} = C_{in}\cdot(\overline{RSQ}+(CR13\cdot CR12\cdot CR11\cdot CR10)\cdot\overline{XB})$$

Additional information on the $\overline{XB}$ line is found in the following Branch Control and Applications sections.

RTN — Return from Subroutine

Tha RTN is an unconditional return from subroutina in which tha LIFO stack is "popped" and the contents of CR4 ara transfarred to CR0. Up to 4 levels of nesting are possible with tha on-chip stack.

The RTN instruction is used with the JSR instruction for normal subroutining or multipla executions, again dependent on tha RSR flip-flop (see Table 2).

1. If RSQ = 0, a normal return is executed. The stack is "popped" and tha contents of CR4 ara loaded into CR0.

2. If RSQ = 1, the stack is "popped" to CR0 and CR1 is incremented. The RTN will continue in the repeat mode until CR1 is filled with all ones. The RSR flip-flop is reset when all CR1 registers reach full count. As with the JSR command $\overline{XB}$ interconnects parallel MC10801's to determine full count.

RSR — Repeat Subroutine

The RSR command initializes the RSR flip-flop and CR1 for repeating microinstructions or subroutines. During the RSR, CR0 is incremented to the next address location (CR0 plus $C_{in} \rightarrow$ CR0), CR1 is loaded from the N.A. inputs, and the RSR flip-flop is set to a logic "1".

Register CR1 determines the number of times a microinstruction or subroutine will be repeated. Used as a cycle counter, CR1 is incremented until the register contains all ones (final count). For this reason, the repeat count originally loaded into CR1 must be the *2's complement* of the desired count number.

Setting the RSR flip-flop to a logic "1" causes JSR and RTN to repeat subroutines and RPI to repeat single microinstructions.

RPI — Repeat Instruction

The RPI command is used to repeat single microinstructions. In a repeat mode (RSR flip flop set to logic 1 by an RSR instruction), RPI holds the CR0 control memory address constant and increments the CR1 repeat counter. At the final repeat count, all "1"s in CR1, the RSR flip-flop is reset to logic "0" and RPI loads the contents of CR1 ANDed with the N.A. inputs into CR0.

The RPI therefore directly jumps to the new address on the N.A. inputs after the microinstruction repeat sequence is complete. (Note that CR1 remains at all "1"s after completing the repeat sequence.)

$\overline{XB}$ is common to all parallel MC10801s to insure CR1 is full on all circuits.

When not in a repeat mode (RSR flip-flop at logic "0"), the RPI instruction becomes a direct jump to register CR1. CR1 is ANDed with the N.A. inputs and loaded into CR0. In this mode RPI is used with JLA for a single level subroutine, where the return address is: (the starting eddress plus C_{in}) ANDed with N.A.

BRC — Branch on Condition

The BRC instruction is a conditional jump to the N.A. inputs. The branch decision is determined by the equation:

$$\overline{XB} \cdot (CS4 + \overline{B})$$

where $\overline{XB}$ is the external Extender Bus common to all parallel circuitry end B is the brench input to any MC10801. If the brench equation equals "0", BRC executes a direct jump to N.A. inputs. If the branch equation equals a logic "1", the present control memory address in CR0 is incremented (CR0 plus C_{in} → CR0) and the progrem goes to the next sequential location.

Normally the test bit is applied to en MC10801 branch, B, input. For multiple chip configurations, the $\overline{XB}$ line is connected common so all MC10801 respond to the same brench signal. Select line CS4 is an enable for the B input and selects which MC10801 B input is tested for the brench decision. A selected CR3 bit may also be used for brenching as described in Table 4.

BSR — Branch to Subroutine

The BSR is a conditional jump to subroutine. The branch condition is determined by the $\overline{XB}$ line and the B input as with BRC. If $\overline{XB} \cdot (CS4 + \overline{B})$ = logic "0" the BSR jumps to subroutine. The subroutine destination on the N.A. inputs is loaded into CR0, and the present address in CR0 is incremented and pushed into the LIFO stack (CR0 plus C_{in} → CR4). If the branch equation equals logic "1", the present control memory address is incremented (CR0 plus C_{in} → CR0).

Unlike JSR, the BSR command is unaffected by the RSR flip flop status. Therefore, a BSR subroutine can be nested within a JSR/RTN repeat subroutine sequence without incrementing the CR1 cycle count register. A ROC is then used to return from the BSR jump.

ROC — Return on Condition

The ROC is a conditional return from subroutine. If the branch equation $\overline{XB} \cdot (CS4 + \overline{B}) = 0$, the return is executed by popping the LIFO stack and loading CR4 into CR0. If the equation equals e logic "1", the MC10801 performs a direct jump in the subroutine by loading the N.A. inputs into CR0. ROC operates independent of the RSR flip flop and can be used with BSR to nest subroutines within a repeat sequence.

BRM — Branch and Modify

The BRM instruction is a jump to the N.A. inputs with an address modification by the B and $\overline{XB}$ inputs. The following information is loaded into CR0 with CR03 being the most significant bit in the part.

CR03 = NA3
CR02 = NA2
CR01 = NA1·XB
CR00 = NA0·B

Note that $\overline{XB}$ is inverted as a modifier. This address modification allows multiway branching where the branches are sequential locations.

CS4 overrides the branch modifiers as shown in Table 2. When multiple MC10801s are operated in parallel, CS4 can be used to disable B and $\overline{XB}$ on all but the two least significant address bits.

REPEAT AND BRANCH CONTROL B, $\overline{XB}$, CS4

The Branch (B), Extender Bus ($\overline{XB}$) and Select line CS4 control certain MC10801 instructions to make repeat or conditional jump decisions.

Brench-B and Select line—CS4:

Brench operations BRC, BSR, and ROC use the B input as a source of decision information. Parallel MC10801 circuits determine the branch status from any B input enabled by select line CS4. The selected B input is routed to the $\overline{XB}$ line which is common to all MC10801s and allows perellel circuits to operate as a unit.

A branch decision depends on the following equation: $\overline{XB} \cdot (CS4 + \overline{B})$. Select line CS4 enables the B input when held at a negative logic "0" (MECL V_{OH}). Branch then occurs if B = logic "1", and the $\overline{XB}$ line extends this branch condition to all parallel circuits. $\overline{XB}$ is a complemented signal to operete properly when wired together using the emitter dot (negative logic AND function).

The BRM instruction is e speciel type of brehch where the B end $\overline{XB}$ lines determine register CR0 bits as follows:

CR03 = NA3
CR02 = NA2
CR01 = NA1•XB
CR00 = NA0•B

Select line CS4 overrides this use of the branch inputs. The ebove CR0 inputs are meintained with CS4 = logic "0" and the 4 NA inputs are routed directly to CR0 when CS4 = logic "1". This feature is used with parallel MC10801 circuits to perform a 4-way branch with the two least significant address bits. CS4 disables branch on the more significant circuits. If two MC10801s are used with CS4 = "0" on both chips, B wey brenching is possible with the next microprogram address being NA7 NA6 NA5•$\overline{XB}$ NA4•B2 NA3 NA2 NA1•$\overline{XB}$ NA0•B1 where B1 is the brench input on the lower order chip and B2 is the brench input on the upper order chip.

Repeat operations JSR, RTN, and RPI respond to $\overline{XB}$, but not to B. The functional equation for a repeet decision is $\overline{RSQ}$ + RIN•$\overline{XB}$. Repeat operation is discussed in the preceding MC10801 instruction descriptions end the following applications section.

Extender Bus — $\overline{XB}$:

The $\overline{XB}$ line operetes in severel modes end cen be driven from verious parts of the MC10801 or from externel circuitry.

The $\overline{XB}$ line is controlled by the B input to insure brench coupling between parallel circuits as described ebove. Status register CR3 bits can be multiplexed onto $\overline{XB}$ with select lines CS0 through CS3. To make branch decisions, the selected CR3 bit goes to ell parellel MC10801s on the $\overline{XB}$ interconnection. Select lines CS0 through CS3 operete independently of the selected MC10801 IC0-IC3 control instruction end must be programmed for the branch.

Repeat register CR1 end the RSR flip-flop control $\overline{XB}$ during a JSR, RTN, or RPI instruction. If RSQ = logic "1" (the MC10801 in a repeat mode) end CR1 signifies a repeat count, $\overline{XB}$ is forced to e logic "0". $\overline{XB}$ going to all parellel MC10801 circuits, couples the cycle count information in CR1 to control the repeat sequence. During a repeat sequence CR3 status bits should be disabled from $\overline{XB}$ to avoid overriding the CR1 cycle count. In e nonrepeat mode, RSQ = logic "0", the $\overline{XB}$ line has no effect on JSR, RTN or RPI instructions.

It is possible to control or modify the $\overline{XB}$ line from an externel signal. The $\overline{XB}$ pins of parallel MC10801s are emitter dotted and en externel signel can be tied into this connection. The external signel would override internel MC10801 control by forcing a negative logic "0" (MECL V_{OH}) on the $\overline{XB}$ line. This feeture is not required for normel MC10801 operetion end would be used to produce speciel branch functions.

Teble 3 is e listing of the $\overline{XB}$ stetus as controlled by the verious MC10801 control sequence instructions and select lines CS0, CS1, CS3 end CS4.

TAeLE 3

TRUTH TAeLE FOR THE $\overline{XB}$ (EXTENOER eUS) LINE

	1			2			3
COMMENTS	REPEAT FUNCTION	BRANCH OISABLE CS4	INSTRUCTION CONTROL IC3-IC0 MNEMONIC CODE	CS3	CS1	CS0	$\overline{XB}$
Brench input or repeat function cannot effect the $\overline{XB}$ line on these instructions	X	X	$\overline{\text{JSR+RPI+RTN+}}\cdots$ $\overline{\text{BRC+BSR+ROC}}$	1 0 0 0 0	X 0 0 1 1	X 0 1 0 1	1 $\overline{CR30}$ $\overline{CR31}$ $\overline{CR32}$ $\overline{CR33}$
Branch input cannot effect the $\overline{Xe}$ line when CS4=1	X	1	BRC+BSR+ROC	1 0 0 0 0	X 0 0 1 1	X 0 1 0 1	1 $\overline{CR30}$ $\overline{CR31}$ $\overline{CR32}$ $\overline{CR33}$
The Brench input is selected onto the $\overline{XB}$ line when CS4=0 and the instruction is a BRC, BSR or ROC	X	0	BRC+BSR+ROC	1 0 0 0 0	X 0 0 1 1	X 0 1 0 1	$\overline{B}$ $\overline{B}\cdot\overline{CR30}$ $\overline{B}\cdot\overline{CR31}$ $\overline{B}\cdot\overline{CR32}$ $\overline{B}\cdot\overline{CR33}$
If the repeat function =0, the $\overline{XB}$ line is unaffected by JSR, RPI, or RTN	0	X	JSR+RPI+RTN	1 0 0 0 0	X 0 0 1 1	X 0 1 0 1	1 $\overline{CR30}$ $\overline{CR31}$ $\overline{CR32}$ $\overline{CR33}$
If the repeat function =1, $\overline{XB}$ is forced to 0 on e JSR, RPI or RTN.	1	X	JSR+RPI+RTN	X	X	X	0

"X" represents e Don't Care Condition

NOTES 1. (RSQ)•$\overline{(\text{CR13}\cdot\text{CR12}\cdot\text{CR11}\cdot\text{CR10})}$ = Repeat Function
2. CS3 enables a bit from CR3 to be placed on $\overline{XB}$, CS0 and CS1 select the bit from CR3.
3. The $\overline{XB}$ line can be forced to a "0" from an external chip using the negative logic "ANO".

STATUS REGISTER CR3 CONTROL CS0, CS1, CS2, CS3

Register CR3 is primarily used as a storage area for microprogram status information. The contents of this register are continuously available on MC10801 package pins CR30 through CR33. Information can be loaded from the I Bus port, NA inputs, or from the single line input, D_{in}. Select lines CS0 through CS3 and the reset, $\overline{RST}$, input control all CR3 load operations. In addition, CS0, CS1, and CS3 enable CR3 bits onto the $\overline{XB}$ line as described in the preceeding section and Table 3.

CS0 and CS1 select one of the four CR3 bits to be loaded from information on the D_{in} input. This occurs with CS2 = logic "0". CS0 and CSI also select the I Bus or NA inputs for parallel loading CR3. Table 4 shows the truth table for entering information into CR3. As with all MC10801 registers, CR3 is a master-slave design which loads information on a positive going (V_{OL} to V_{OH}) clock edge.

TABLE 4

TRUTH TABLE FOR
STATUS REGISTER CR3 AND $\overline{XB}$ AS A FUNCTION OF CS0-CS3

$\overline{RST}$	SELECT LINE INPUTS CS3	CS2	CS1	CS0	REGISTER CR3 OUTPUTS CR33	CR32	CR31	CR30	$\overline{XB}$
0	X	X	X	X	0	0	0	0	–
1	0	0	0	0	–	–	–	DIN	$\overline{CR30}$
1	0	0	0	1	–	–	DIN	–	$\overline{CR31}$
1	0	0	1	0	–	DIN	–	–	$\overline{CR32}$
1	0	0	1	1	DIN	–	–	–	$\overline{CR33}$
1	0	1	0	0	–	–	–	–	$\overline{CR30}$
1	0	1	0	1	–	–	–	–	$\overline{CR31}$
1	0	1	1	0	–	–	–	–	$\overline{CR32}$
1	0	1	1	1	–	–	–	–	$\overline{CR33}$
1	1	0	0	0	–	–	–	DIN	1
1	1	0	0	1	–	–	DIN	–	1
1	1	0	1	0	–	DIN	–	–	1
1	1	0	1	1	DIN	–	–	–	1
1	1	1	0	0	0	0	0	0	1
1	1	1	0	1	IB3	IB2	IB1	IB0	1
1	1	1	1	0	NA3	NA2	NA1	NA0	1
1	1	1	1	1	–	–	–	–	1

"X" represents a Don't Care Condition; "–" represents a NO CHANGE Condition

NOTES 1. Register CR3 changes stata on a V_{OL} to V_{OH} transition at the clock input.

2. The $\overline{XB}$ line can be forced to a "0" due to a branch or repeat condition. Table 3 fully describes $\overline{XB}$.

CR0 OUTPUT BUFFER ENABLE CS5

Select line CS5 provides a gating function on the CR0 control memory address outputs. A logic "1" on CS5 enables CR0 to package pins CR00 through CR03. A logic "0" on CS5 forces the buffer outputs to a logic "1" state. This negative logic 1 (MECL V_{OL}) frees the CR0 output pins and allows for an external source of control memory address information. Note that when the CR0 buffers are disabled, the CR0 information is still available for internal operation. This alternate addressing feature can be used to load writable control storage on power up or for forcing interrupt vectors and overriding normal MC10801 operation. Table 5 shows the truth table for the CS5 input.

TABLE 5

TRUTH TABLE FOR CR0 OUTPUT BUFFER

CS5	OUTPUTS CR00 – CR03
1	ENABLED
0	DISABLED

BUS CONTROL CS6, CS7, CS8

The I Bus and Ø Bus function as I/O ports for information stored within the MC10801 internal registers. For data output, CS6, CS7, and CSB select the proper register and enable the bus output drivers. When not used to output data the MC10801 internal bus drivers are forced to a negative logic 1 (MECL V_{OL}) to provide for I Bus and Ø Bus data input operations.

Lines CS6, CS7, and CSB select data from registers CR1, CR2, or either end of the LIFO stack CR4 and CR7. CS6 selects either the I Bus or the Ø Bus while CS7 and CSB control the source of output data. Registers CR1 and CR2 are directly selected. However, CR4 and CR7 selection is dependent upon IC0-IC3 control instructions involving the LIFO. CR7 can be read only during a JSR or BSR with branch LIFO push operation. Reset, $\overline{RST}$, results in a LIFO push and also enables CR7 as an output.

LIFO pop operations, as caused by a RTN or ROC with branch, forces a logic 1 state on the I Bus and Ø Bus drivers. Either port can then input information to CR7 as required to extend the stack depth with external circuits. All MC10801 control instructions not involving the LIFO enable CR4 as a possible I Bus or Ø Bus data source. Table 6 shows registers available to the I Bus and Ø Bus as output ports.

TABLE 6

SELECTING THE I BUS AND Ø BUS AS DATA OUTPUTS

INSTRUCTION CONTROL IC0 – IC3 MNEMONIC CODE	$\overline{RST}$	CS7	CS8	CS6 = 0 ØB	IB	CS6 = 1 ØB	IB
X	X	0	0	1	CR1	CR1	1
JSR+BSR·XB	X	0	1	1	CR7	CR7	1
X	0	0	1	1	CR7	CR7	1
RTN+ROC·XB	1	0	1	1	1	1	1
$\overline{\text{JSR+RTN+(BSR+ROC)·XB}}$	1	0	1	1	CR4	CR4	1
X	X	1	0	1	CR2	CR2	1
X	X	1	1	1	1	1	1

X = Don't care

The bus control inputs also select either the I Bus or Ø Bus as input ports to load information into the bottom of the LIFO (CR7). Select line CS6 selects either the I Bus or Ø Bus while CS7 and CSB in conjunction with a LIFO pop function, RTN or ROC with branch, enables these ports as inputs to CR7. Table 7 shows complete LIFO operation and selection of I Bus and Ø Bus as controlled by the IC0-IC3 control instructions and the CS6-CSB bus control inputs.

The I Bus automatically becomes an input port to CR0 or CR2 during a JIB, JIN, or JL2 instruction. When using these instructi· ,.s a logic "1" is normally selected on the I Bus drivers, see Table 6, to avoid a conflict between internal register data and the incoming I Bus information.

CARRY OUT C_{OUT}

The C_{out} line is a direct function of the C_{in} input and the CR1 or CR0 registers as shown in Table B. Note that when RSQ = 0, C_{out} always monitors the CR0 register independent of the IC0—IC3 instruction inputs.

TABLE 7

TRUTH TABLE FOR THE 4 X 4 LIFO STACK (REGISTERS CR4-CR7)

INSTRUCTION CONTROL ICO – IC3 MNENONIC CODE							NEXT STATE			
	$\overline{\text{RST}}$	$\overline{\text{XB}}$	RSO	CS6	CS7	CS8	CR4	CR5	CR6	CR7
' RTN + RPI	0	X	1	X	X	X	CR1	CR4	CR5	CR6
RTN + RPI	0	X	0	X	X	X	CR0	CR4	CR5	CR6
$\overline{\text{RTN + RPI}}$	0	X	X	X	X	X	CR0	CR4	CR5	CR6
JSR	1	X	0	X	X	X	CR0 Plus C_{IN}	CR4	CR5	CR6
JSR	1	1	1	X	X	X	CR0 Plus C_{IN}	CR4	CR5	CR6
JSR	1	0	1	X	X	X	CR0	CR4	CR5	CR6
BSR	1	1	X	X	X	X	–	–	–	–
BSR	1	0	X	X	X	X	CR0 Plus C_{IN}	CR4	CR5	CR6
RTN	1	X	X	X	0	0	CR5	CR6	CR7	0
RTN	1	X	X	X	1	X	CR5	CR6	CR7	0
RTN	1	X	X	0	0	1	CR5	CR6	CR7	IB
RTN	1	X	X	1	0	1	CR5	CR6	CR7	Ø0
ROC	1	I	X	X	X	X	–	–	–	–
ROC	1	0	X	X	0	0	CR5	CR6	CR7	0
ROC	1	0	X	X	1	X	CR5	CR6	CR7	0
ROC	1	0	X	0	0	1	CR5	CR6	CR7	IB
ROC	1	0	X	1	0	1	CR5	CR6	CR7	ØB
$\overline{\text{JSR+BSR+RTN+ROC}}$	1	X	X	X	X	X	–	–	–	–

"X" represents a Don't Care Condition, "–" represents a NO CHANGE Condition

TABLE 8

TRUTH TABLE FOR C_{out}

INSTRUCTION CONTROL ICO – IC3	RSQ	C_{out}
RPI + RTN	0	C_{in}·CR03·CR02·CR01·CR00
RPI + RTN	1	C_{in}·CR13·CR12·CR11·CR10
$\overline{\text{RPI + RTN}}$	X	C_{in}·CR03·CR02·CR01·CR00

APPLICATIONS INFORMATION

The MC10801 fits a wide range of system sizes and applications, and therefore, has no fixed interconnection configuration. The specific system design goals will determine the control memory size, the number of MC10801s, and the interconnection pattern. A typical small processor control section can, however, illustrate use of the MC10801. Figure 3 shows two MC10801s plus microprogram control storage for the processor. Various features are described below:

MEMORY ADDRESSING

Two MC10801s provide increment, direct jump, branch, and subroutine capability for up to 256 words of control memory. Three devices can extend this to 4K words. Control register CR0 outputs are the control memory address.

A second technique to extend memory addressing beyond 256 words is two MC10801s and word-page memory mapping. Status Register CR3 of device B extends the memory size to 16 pages of 256 words each. Increment, direct jump, branch, and subrouting are restricted to within a given page, however, the third MC10801 and Next Address feedback bits from control storage are eliminated. The page address is loaded from the I 8us or NA inputs and controlled via the Status Field.

CONTROL STORAGE

Control storage can be as large as 4K words (16 pages x 256 words) for the example shown. If writable control storage is desired, MECL RAM's (MCM 10144 or MCM 10146) are used. For PROM the MCM 10149 is used.

The word length is the sum of the various control fields existing in the control storage. The Instruction Field equals 4 bits, the Next Address Field equals 8 bits, the Status Field is up to 10 bits, etc. It is not unusual for the word size to be 40 to 80 bits or more including the RF, ALU condition code, and other processor fields.

If system cycle times permit, the word size can be decreased by control field decoding. Small PROMs such as the MCM10139 or discrete logic are used to decode the select line signals. The number of microprogram bits can be reduced, but additional delay in the feedback path is introduced.

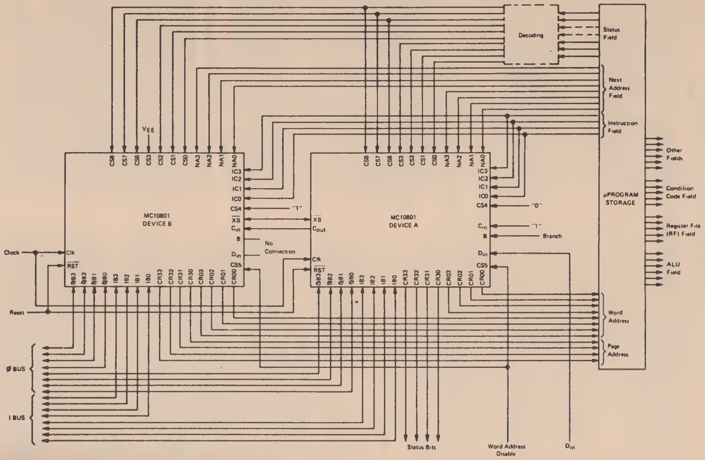
MICROPROCESSOR SEQUENCE CONTROL

The control fields feedback from storage to the MC10801s determine the microprogram sequence. The 4-bit Instruction Field selects one of 16 control instructions to generate the next microprogram address. Instruction lines ICO through IC3 are respectively tied in parallel so that devices A and 8 perform the same instruction.

The Next Address Field is 8 bits wide – four bits to the most significant device 8 and the other four bits to device A. The NA inputs are the source of constants, starting addresses, jump and branch vectors, subroutine vectors, and masking information. The data at the NA inputs is used by the MC 10801 and controlled by the Instruction Field and/or the Status Field.

The Status Field can be up to 10 bits wide. The number of bits can be decreased with decoding or with selective functions used in the MC10801.

FIGURE 3 – PROGRAMMED CONTROL



Control lines CS0 through CS3 are driven independently because they manipulate register CR3 on each MC10801. Register CR3 is the Page Address register on device B, and is a status bit register on device A. Each CR3 register on the two MC10801's must be controlled independently.

Other connections to the MC10801s include:

1. C_{in} of the least significant device A is a logic "1" for increment functions. C_{out} of device A ripples to C_{in} of device B.

2. The $\overline{XB}$ lines are tied common for parallel branch functions.

3. Branch information is tested on the B input of device A. As a result, CS4 = 1 for device B to disable its branch input because the input is not used.

4. Clock and reset are tied in parallel on both devices.

5. CS5 is the CR0 output disable or the Word Address Disable. This line can be used for writable control storage functions or for interrupt functions.

6. Data can be entered into the CR3 register on a single bit basis using the D_{in} input.

I BUS AND Ø BUS

The data buses are tied to other ports of the processor. Starting addresses, interrupt vectors, and extension of the internal LIFO stack are common uses of these buses. Both the I Bus and Ø Bus are bidirectional, and are controlled by the Status Field and the Instruction Field.

SUBROUTINE & REPEAT FUNCTIONS

Subroutine and repeat operations are important functions of the MC 10801. These can each be illustrated simply.

1. Non-repeat subroutine — an example is illustrated in Figure 4. The address is limited to the word address and is listed in hexadecimal. At address 06, a JSR is executed in which address 17 → CR0, the present address plus 1 → CR4, and the stack is pushed. The subroutine begins at address 17 and ends at address 19 with an RTN. When the RTN is executed, (CR4) → CR0, the stack is popped, and the program jumps to location 07 to continue the program.

FIGURE 4 – NON-REPEAT SUBROUTINE LISTING

ADDRESS	I FIELD	NA FIELD	DESCRIPTION
06	JSR	17	06+1 → CR4, Push Stack, 17 → CRO
07	INC	X	Continue Program
–			
–			
–			
17	INC	X	Beginning of Subroutine
18	INC	X	
19	RTN	X	(CR4) → CR0, Pop Stack

2. Repeat subroutine — Figure 5 shows this example. The instruction flow is similar to the above example except that an RSR must be executed. During the RSR, CR1 is loaded with the 2's complement of 4 which is the number of times the subroutine is to be repeated. In hexadecimal notation, this is FC for 4 cycles. Also RSQ is set to 1 for repeat (in the non-repeat mode RSQ = 0).

The JSR is executed to begin the subroutine operation. During the JSR, the subroutine address 17 → CR0, the present address 06 plus C_{in} internal → CR4 and the stack is pushed. If RSQ = 0 or CR1 = FF, C_{in} internal = 1. Thus for the first 3 cycles when the JSR is executed, the present address 06 is loaded into CR4.

At the end of each subroutine cycle, an RTN is executed or (CR4) → CR0; the stack is pushed; and if RSQ = 1 and CR1 ≠ FF, then CR1 is incremented and if CR1 = FF, 0 → RSQ. In this example for the first 3 cycles, the RTN jumps to 06 (the JSR) and CR1 is incremented finally to FF.

On the final cycle, the JSR is executed with CR1 = FF and address 07 is loaded into CR4. Then, the RTN resets RSQ and jumps to location 07 to end the operation.

For this example with an 8-bit word address, the maximum number of subroutine cycles is 256.

FIGURE 5 – REPEAT SUBROUTINE LISTING

ADDRESS	I FIELD	NA FIELD	DESCRIPTION
05	RSR	FC	1111 1100 → CR1, 1 → RSQ
06	JSR	17	06 + C_{IN} Internal → CR4 Push Stack 17 → CR0 If RSQ = 0 or CR1 = FF, then C_{IN} Internal = 1
07	INC	X	Continue Program
–			
–			
17	INC	X	Beginning of Subroutine
18	INC	X	
19	RTN	X	(CR4) → CR0, Pop Stack if RSQ = 1 and CR1 ≠ FF then (CR1) +1 → CR1, if CR1 = FF, then 0 → RSQ

3. Repeat Instruction is shown in Figure 6. As in the repeat subroutine, an RSR is executed loading CR1 with = FC and RSQ is set. This sets the number of instruction cycles at 4.

   An RPI then is executed. If RSQ = 1 and CR1≠ FF, then (CR1) plus 1 → CR1 and CR0 → CR0. If CR1 = FF, RSQ is reset and (11)•(FF) → CR0.

   Thus for the first 3 cycles, CR1 is incremented and CR0 stays at the present address.

   During the fourth and final cycle (CR1 = FF from the third cycle) RSQ is reset and CR0 jumps to the Next Address value (11) AND'ed with the value of CR1 (FF) which is all logic 1s. The location 11 now continues the program.

   The maximum repeat cycle is again 256.

FIGURE 6 – REPEAT INSTRUCTION LISTING

ADDRESS	I FIELD	NA FIELD	DESCRIPTION
09	RSR	FC	1111 1100 → CR1, 1 → RSQ
0A	RPI	11	If RSQ = 1 and CR1 ≠ FF, then (CR1) + 1 → CR1, (CR0) → CR0. If CR1 = FF, then 0 → RSQ (11·FF) → CR0
11	INC		Continue Program

RECOMMENDED OPERATING CONDITIONS

· PARAMETER	SYMBOL	VALUE	UNIT
Supply Voltage (V_{CC} = 0 Volts)	V_{TT} V_{EE}	−1.9 to −2.2 −4.68 to −5.72	Vdc Vdc
Operating Temp. (Functional)	T_A	−30 to +85	°C
Output Drive	–	50Ω to −2.0 Vdc	–
Maximum Clock Input Rise and Fall Time (20% to 80%)	t_r, t_f	10	ns
Minimum Clock Pulse Width	PW	5	ns

ELECTRICAL CHARACTERISTICS

Each MECL 10,000 series circuit has been designed to meet the dc specifications shown in the test table, after thermal equilibrium has been established. The circuit is in a test socket or mounted on a printed circuit board and transverse air flow greater than 500 linear fpm is maintained. Outputs are terminated through a 50-ohm resistor to −2.0 volts. Test procedures are shown for only one input, or for one set of input conditions. Other inputs tested in the same manner.

@Test Temperature	TEST VOLTAGE VALUES					
	Volts					
	V_{IHmax}	V_{ILmin}	V_{IHAmin}	V_{ILAmax}	V_{EE}	V_{TT}
−30°C	−0.890	−1.890	−1.205	−1.500	−5.2	−2.0
+25°C	−0.510	−1.85	−1.105	−1.475	−5.2	−2.0
+85°C	−0.700	−1.825	−1.035	−1.440	−5.2	−2.0

Characteristic	Symbol	Pin Under Test	TEST LIMITS								VOLTAGE APPLIED TO PINS LISTED BELOW						(V_{CC}) Gnd
			−30°C		+25°C			+85°C									
			Min	Max	Min	Typ	Max	Min	Max	Unit	V_{IHmax}	V_{ILmin}	V_{IHAmin}	V_{ILAmax}	V_{EE}	V_{TT}	
Power Supply Drain Current	I_{EE}	1, 24	–	–	–	200	–	–	–	mAdc	–	–	–	–	1, 24	25, 48	7,12,17,36
	I_{TT}	25, 48	–	–	–	270	–	–	–						1, 24	25, 48	7,12,17,36
Input Current	I_{inH}	23	–	–	–	–	45	–	–	μAdc	23	–	–	–	1, 24	25, 48	7,12,17,36
		42	–	–	–	–	370	–	–	↓	42				↓	↓	↓
		40	–	–	–	–	470	–	–		40						
	I_{inL}	18	–	–	0.5	–	–	–	–		–	18	–	–			
Logic "0" Output Voltage	V_{OH}	16	−1.060	−0.890	−0.960		−0.810	−0.890	−0.700	Vdc	*16,26,27,45	–	–	–	1, 24	25, 48	7,12,17,36
		2	−1.060	−0.890	−0.960	–	−0.810	−0.890	−0.700	Vdc	41,44,46	40,42,43	–	–	1, 24	25, 48	7,12,17,36
Logic "1"† Output Voltage	V_{OL}	18	−1.940	−1.675	−1.900	–	−1.850	−1.875	−1.515	Vdc	45	26,27,40	–	–	1, 24	25, 48	7,12,17,36
		2	−1.890	−1.675	−1.850	–	−1.650	−1.825	−1.615	Vdc	**42,43,45	40,41,44,46,47	–	–	1, 24	25, 48	7,12,17,36
Logic "0" Threshold Voltage	V_{OHA}	2	−1.080	-	−0.980	–	–	−0.910	–	Vdc	**42,43,45	40,41,44,47	46	–	1, 24	25, 48	7,12,17,36
Logic "1" Threshold Voltage	V_{OLA}	2	–	−1.655	–	–	−1.630	–	−1.595	Vdc	**42,43,45	40,41,44,47	–	46	1, 24	25, 48	7,12,17,36

†The bi-directional outputs are specified at −1.90 volts for V_{OL} min

Preset Conditions

*PS1 Apply V_{IH} at 37, 43 V_{IL} at 40, 41, 42, 44, then clock once (⊔).

**PS2 Apply V_{IH} at 41, 42, 43, 44. V_{IL} at 33. 34. 35, 37, 40, 47: then clock once (⊔)

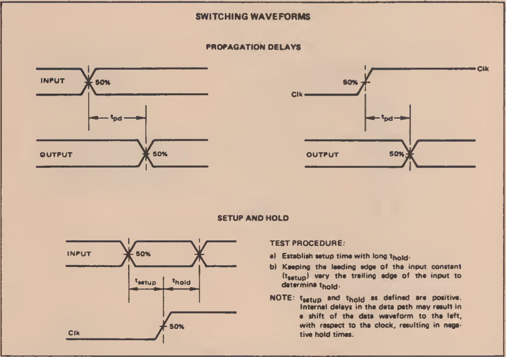
SETUP AND HOLD TIMES
(NANOSECONDS AT 25°C)

Input	Setup Min	Hold Min
ICO–IC2 (1)	27	−2.0
ICO–IC3 (2, 3)	44	−8.0
NA0–NA3	28	+2.0
I Bus, Ø Bus	25	+1.0
CS0–CS3	35	−2.0
CS4 (4)	23	−2.0
B (4)	21	−1.0
C_{in}	15	+2.0
D_{in}	20	+2.0
$\overline{RST}$	20	+5.0
$\overline{XB}$	28	−4.0
$\overline{XB}$ (4)	20	−2.0

NOTES:
(1) All instructions except 2 and 3 below.
(2) BSR, BRC, BRM, or ROC instruction when B · $\overline{CS4}$ = 1.
(3) BSR, BRC, BRM, ROC, JSR, RPI, or RTN instruction when RSQ = 1.
(4) BRM instruction only.

PROPAGATION DELAY TIMES (NANOSECONDS)

		−30°C		+25°C		+85°C	
Input	Output	Typ	Max	Typ	Max	Typ	Max
Clock	CR0, CR3	11.1	16.0	11.2	16.0	12.0	17.0
Clock	IB, ØB	16.3	30.0	16.8	30.0	21.2	32.0
Clock	$\overline{XB}$	15.7	20.0	16.1	21.0	18.2	23.0
Clock	C_{out}	13.5	22.0	14.6	23.0	16.3	24.0
C_{in}	C_{out}	3.10	9.00	2.80	7.00	3.40	8.00
ICO–IC3	ØB	22.7	32.0	23.4	33.0	28.9	38.0
ICO–IC3	$\overline{XB}$	14.9	20.0	15.9	21.0	19.1	24.0
ICO–IC3	C_{out}	17.4	26.0	17.4	26.0	20.9	27.0
CS7, CS8	IB, ØB	14.8	22.0	16.0	24.0	17.6	28.0
CS0–CS4, B	$\overline{XB}$	11.8	17.0	12.4	18.0	15.3	20.0
CS6	IB, ØB	7.00	11.0	6.80	11.0	7.70	12.0
CS5	CR0	5.20	10.0	5.30	10.0	6.10	11.0
$\overline{XB}$	IB, ØB	21.3	29.0	22.2	31.0	24.6	36.0
$\overline{RST}$	IB, ØB	18.8	26.0	19.6	28.0	22.9	31.0
TR, TF	All	6.00	11.0	6.50	11.0	8.30	12.0

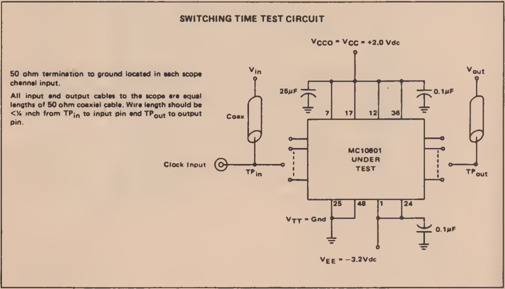


TEST PROCEDURE:

a) Establish setup time with long t_{hold}.

b) Keeping the leading edge of the input constant (t_{setup}) vary the trailing edge of the input to determine t_{hold}.

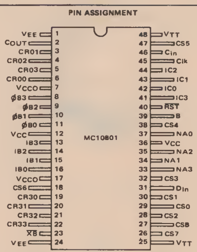
NOTE: t_{setup} and t_{hold} as defined are positive. Internal delays in the data path may result in a shift of the data waveform to the left, with respect to the clock, resulting in negative hold times.



50 ohm termination to ground located in each scope channel input.

All input and output cables to the scope are equal lengths of 50 ohm coaxial cable. Wire length should be <¼ inch from TP_{in} to input pin and TP_{out} to output pin.

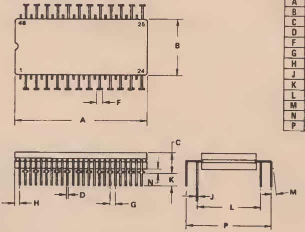
PIN ASSIGNMENT



THERMAL CHARACTERISTICS
(TYPICAL)
@ 500 Linear Ft. Air Flow

θ_{JA} = 26.5° C/W
θ_{JC} = 7° C/W

PACKAGE DIMENSIONS



	MILLIMETERS		INCHES	
DIM	MIN	MAX	MIN	MAX
A	31.24	32 26	1 230	1.270
B	12 70	13.72	0.500	0.540
C	4 57	5.59	0.180	0 220
D	0 38	0.53	0.015	0.021
F	1.14	1.40	0.045	0.055
G	1 27 BSC		0.050 BSC	
H	1.02	1.52	0 040	0.060
J	0.20	0.30	0.008	0.012
K	2.54	3.30	0.100	0.130
L	15.24 BSC		0.600 BSC	
M	–	7°	–	7°
N	0.51	1.52	0.020	0.060
P	20.32 BSC		0.800 BSC	

Case 725-01

A socket for the QUIL package is available from ELECTRONIC MOLDING CORPORATION. (Part number 7178-295-6)

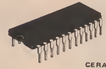
QUIL is a trademark of Motorola Inc.

MOTOROLA

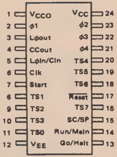
MC10802

Advance Information

MECL – LSI
TIMING
FUNCTION



L SUFFIX
CERAMIC PACKAGE
CASE 623



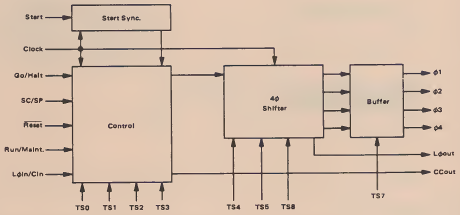
INTRODUCTION

The MC10802 Timing Function is an LSI building block for digital processor systems. This circuit contains the logic and control lines to generate system clock phases and provides for start, stop, and diagnostic operations. Each part is four bits wide and can be connected in series for greater than four phase clock systems.

The Timing Function as shown in the block diagram below is composed of a four phase shifter circuit with buffered outputs. Fifteen input lines combine with Control and Start Sync logic to control all operations within the part.

FEATURES

- Programmable Number of Phases
- Selectable Double-Width Phases Duration
- Start Signal Synchronizer
- Single Cycle Stepping
- Single Phase Stepping
- Asychronous Master Reset
- Cascadable
- Fully Compatible with the MECL 10,000 Family



This is advance information on a new introduction and specifications are subject to change without notice

MOTOROLA

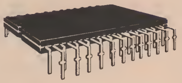
MC10803

INTRODUCTION

The MC10803 Memory Interface Function is an LSI building block for interfacing a high speed processor system to main memory or peripheral equipment. The circuit contains the logic and storage registers for generating memory address and routing incoming or outgoing data. Each part is 4-bits wide and can be connected in parallel to meet wider system I/O word requirements. An internal ALU allows the MC10803 to also assume processor ALU responsibility for many controller applications. Maximum system flexibility is maintained with 5 separate data ports.

The Memory Interface Function as shown in the block diagram below contains six 4-bit registers, an ALU with encoded function/operand select logic, and data transfer circuitry on a single MECL bipolar LSI circuit. Fifteen select (MS) lines control register selection, 13 basic ALU functions, and 17 data transfer operations.

MECL — LSI MEMORY INTERFACE FUNCTION



CASE 725-01

MEMORY INTERFACE FUNCTION
BLOCK DIAGRAM – MC10803

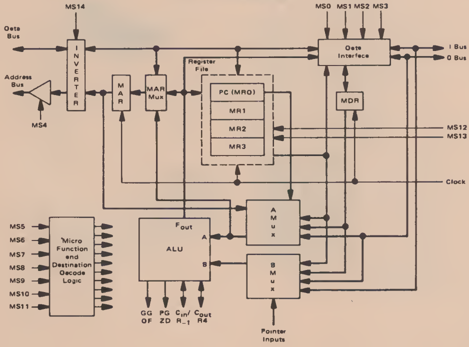
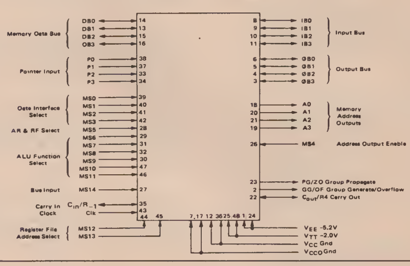


TABLE OF CONTENTS

IMPORTANT FEATURES

1. Internal ALU for address genaration
   a. 13 arithmetic, logic and shift functions
   b. 7 separate ALU operands
2. Four word register fila
   a. Program counter
   b. 3 general purpose registars for index ragistars, stack pointers, etc.
3. Memory data register
4. Mamory address register
5. 17 data transfer and storage operations
6. 4 bits wide and fully expandable
7. 5 data ports for maximum versatility
8. Internal Register File can be expandad by using External Register File connected to tha I Bus and 0 Bus.
9. Fully compatible with MECL 10000
   a. Power supplies
   b. Logic levels

INPUT/OUTPUT DIAGRAM-MC10803



ABSOLUTE MAXIMUM RATINGS (see Note 1)

RATING		SYMBOL	VALUE	UNIT
Supply Voltage (V_{CC} = 0)		V_{EE} V_{TT}	−8 to 0 −4 to 0	Vdc Vdc
Input Voltage (V_{CC} = 0)	Std Bus	V_{in} V_{in}	0 to V_{EE} Note 2	Vdc Vdc
Output Source Current	Cont Surge	I_o I_o	< 50 < 100	mAdc mAdc
Storage Temp.		T_{stg}	−55 to +150	°C
Junction Temp.		T_J	165	°C

NOTE. 1. Permanent device damage may occur if absoluts maximum ratings ara exceeded. Functional operation should be restricted to RECOMMENDEO OPERATING CONOITIONS. Exposure to higher then recommended voltages for extended periods of time could affect device reliability.

NOTE. 2. Input voltage limit is V_{CC} to −2 Volts when tha bus is used as an input and the output drivers are disabled.

MOTOROLA

MC10806

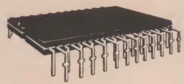
Advance Information

INTRODUCTION

The MC10806 Dual Access Stack is an LSI building block for digital processor systems. This circuit consists of 32 words by 9 bits of memory with two independent address and data ports. The circuit is easily expandable in both the word and bit directions making it ideal in register file, scratch pad, and high-speed buffer applications.

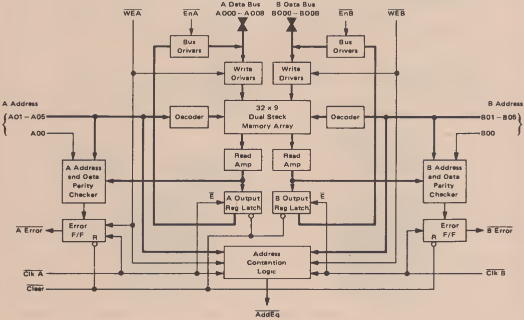
The Dual Access Stack, as shown in the block diagram below, contains a 32 x 9 memory array, two address ports, two 9-bit data input/output ports, two 9-bit output registers, address and data parity checking logic, and two error flip-flops in a single MECL Bipolar LSI circuit. Separate read, write, and output enables exist for each port to control all operations within the part.

MECL — LSI
DUAL ACCESS STACK



CASE 725-01

DUAL ACCESS STACK BLOCK DIAGRAM – MC10806



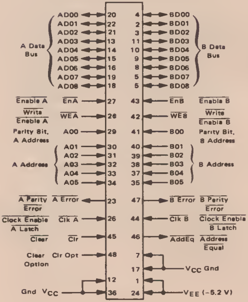
This is advance information on a new introduction and specifications are subject to change without notice.

TABLE OF CONTENTS

IMPORTANT FEATURES

1. 32 x 9 Memory Array
2. Two 9-Bit Output Registers (Latches)
3. Two Independent Address Ports
4. Two Data I/O Ports
5. Address and Data Parity Checking Logic
6. Two Master/Slave Error Flip-Flops
7. Separate Read, Write, and Output Enables for Each Part
8. Each Part is 9-Bits Wide (One Byte) and Can 8e Operated in Parallel to Form Any Word Size in Increments of 9 Bits
9. Fully Compatible with the MECL 10,000 Family

INPUT/OUTPUT DIAGRAM



ABSOLUTE MAXIMUM RATINGS (See Note 1)

Reting		Symbol	Velue	Unit
Supply Voltege (V_{CC} = 0)		V_{EE}	-8 to 0	Vdc
Input Voltege	Std	V_{in}	0 to V_{EE}	Vdc
(V_{CC} = 0)	Bus	V_{in}	Note 2	Vdc
Output Source	Cont	I_{out}	< 50	mAdc
Current	Surge	I_{out}	< 100	mAdc
Storege Temperature		T_{stg}	-55 to +150	°C
Junction Tempareture		T_J	165	°C

NOTES: 1. Permenent device demege mey occur, if ebsolute meximum retings ere exceeded. Functionel operetion should be restricted to RECOMMENDED OPERATING CONDITIONS. Exposure to higher then recommended voltages for extended periods of time could effect device reliebility.

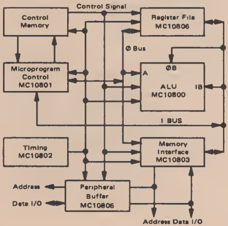
2. Input voltage limit is V_{CC} to -2 Volts when the bus is used es an input and the output drivers ere disabled.

SYSTEM OVERVIEW

The Motorola M10800 family of LSI processor circuits combines the cost and size advantages of LSI with system design flexibility. Each family part is a major system building block which can be interconnected and programmed for a wide range of processor applications. Figure 1 illustrates a method of using the various circuits in a general-purpose processor. The MC10800 4-Bit ALU Slice performs the various arithmetic, logic, and shift functions. This circuit features full BCD capability and a complete set of status outputs. The MC10801 Microprogram Control Function addresses and sequences through microprogram control memory. A set of 16 control instructions provides for direct jumps, conditional branches, and subroutines within microprogram. The MC10802 Timing Function generates clock phases and features single-cycle or single-phase clock increment for troubleshooting or diagnostics.

The Register File has been made a separate block so that the designer can optimize the size and configuration for his particular system. The main function of the Register File is to provide storage for addresses and data. Also, the access time of the Register File must be fast in order to efficiently utilize the high speed of the overall processor system.

FIGURE 1 – MICROPROGRAMMED PROCESSOR



The MC10806 Dual Access Stack provides the register-file function in the processor as well as providing a memory buffer interface to peripheral devices. The MC10806 contains 32 words by 9 bits of memory in which 2 words can be independently addressed for read or write operations on two separate data I/O ports. Also, the circuit has the ability to check for parity errors on both the address and the data.

In the Register File block of the processor configuration shown in Figure 1, the two Data I/O ports of the MC10806 are connected to the two internal busses of the processor, the I Bus and the Ø Bus. The addresses and control signals are connected to the Control Memory. In the configuration shown, two locations of register file can be operated on by the ALU when the clock is at "0" with the result placed on the I Bus when the block is at "1", so that it may be written back into the register file in the same microcycle. More details for this configuration are given in the Application section.

In the Peripheral Buffer block, the MC10806 can be used as a temporary buffer for storing data from the processor to the peripheral device or vice versa. One Data I/O and Address port of the MC10806 is connected to the Data I/O and Address port of the MC10803. The other Data I/O and Address port are controlled by the peripheral device independent of the processor. In this application, the processor can read and write data into the peripheral buffer at high speed using the MC10803, while the peripheral device can read and write into buffer at a slower or faster speed independent of the processor. Flag status and interrupt conditions could also be designed into the peripheral interface depending on the application requirements.

The Motorola M10800 circuits interface directly to all parts in the MECL 10,000 family. This provides a source for high-speed memories and a complete mix of MSI and SSI circuits. Circuits are available for special hardware functions from high-speed multiply to error detection and correction.

Versatility is a key word to describe each circuit in the Motorola M10800 family. The block diagram in Figure 1 and the examples in this data sheet are intended to illustrate ways to use these LSI parts and do not restrict the designer to any particular system configuration or application.

MOTOROLA

MC10808

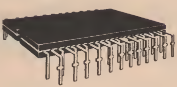
Advance Information

INTRODUCTION

The MC10808 Programmable Multi-Bit Shifter is an LSI building block for shifting data in a high-speed processor system. The circuit is essential when performing floating point operations for pre-normalization or alignment of exponents.

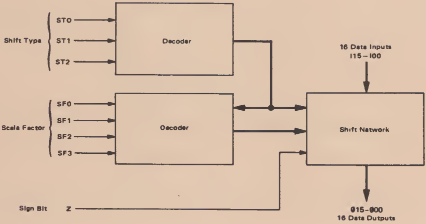
The Programmable Multi-Bit Shifter as shown in the block diagram contains a 16-bit shift network that is fully expandable in a shifter array to handle practically any number of bits. The shift type function select contains arithmetic, logic, and rotate shifting. Four scale factor inputs are provided for specifying the number of positions that the input data is to be shifted or rotated. A sign bit is also provided for arithmetic shift operations.

MECL — LSI PROGRAMMABLE 16-BIT SHIFTER FUNCTION



CASE 725-01

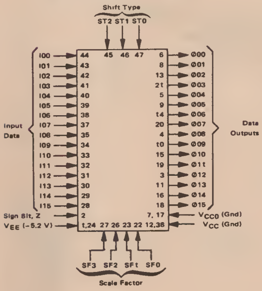
BLOCK DIAGRAM



This is advance information and specifications are subject to change without notice.

TABLE OF CONTENTS

INPUT/DUTPUT DIAGRAM – MC10808



IMPORTANT FEATURES

1. Three hundred gate complexity reduces package count considerably while increasing system speed.
2. Sixteen separate data inputs and sixteen data outputs are available with only two levels of gating separating them for high-speed operation. The sign bit input goes through only one level of gating.
3. Three shift type select lines are used to select eight different shift functions including shift left, shift right, and rotate.
4. Four scale factor inputs select the number of positions (in binary and 2's complement for shift right and shift left) the data is to be shifted.
5. Sign bit input is used for arithmetic shifting and for sign extend operations. Also, the sign bit is used in logic shifting for use in both positive and negative logic systems.
6. The outputs may be disabled for array expansions by selecting the "ODA" function.
7. High-speed operation of 6 ns typ delay from Data-In to Data-Out, 6 ns typ delay from Sign Bit to Data-Out, and 12 ns typ delay from the select lines to the Data-Out.
8. Two different shifter arrays can be built. One array requires only two package delays for a shifter requiring up to 256 bits. The other array requires only one package delay but more packages. A 64-bit shifter requires ten MC10808s with two package delays or sixteen MC10808s with one package delay.
9. Fully compatible with the MECL 10,000 family.

ABSOLUTE MAXIMUM RATINGS (See Note)

Rating	Symbol	Value	Unit
Supply Voltage (V_{CC} = 0)	V_{EE}	–8 to 0	Vdc
Input Voltage (V_{CC} = 0)	V_{in}	0 to V_{EE}	Vdc
Output Source Current Cont Surge	I_o	< 50 < 100	mAdc
Storage Temperature Junction Temperature	T_{stg} T_j	–55 to +150 165	°C

NOTE: Permanent device damage may occur if absolute maximum ratings are exceeded. Functional operation should be restricted to RECOMMENDED OPERATING CONDITIONS. Exposure to higher than recommended voltages for extended periods of time could affect device reliability.